PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

20 AOÛT 1955 UNIVERSITY - SKIKDA

Faculty of Sciences

Department of Computer Science

Dissertation

Submitted in partial fulfillment of the requirements for the Master's degree in Computer Science

Option: Artificial Intelligence

Presented by **Aya KERKAR**

Theme

Transformers-based Approach for Speech Emotion Recognition

Jury:

Chairman	Dr. Hanene MAGROUNE	University 20 août 1955-Skikda
Reviewer	Dr. Khaira TAZIR	University 20 août 1955-Skikda
Supervisor	Dr. Samira HAZMOUNE	University 20 août 1955-Skikda

June 2024

Acknowledgments

Thank Allah the Almighty who has granted me faith, courage, and patience to accomplish this work.

I would like to express my deep gratitude to my supervisor, Dr. Samira HAZMOUNE, for the confidence she placed in me, her constant presence, guidance, humility, advice, and constructive remarks, which greatly contributed to the progress of this work.

I also extend my thanks to the members of the jury who honored me by examining my work, including chairman Dr. Hanene MAGROUNE and Reviewer Dr. Khaira TAZIR.

My deep gratitude goes to my parents, my brother, my sisters, and my entire family for their encouragement, which enabled me to achieve this modest work. I am very grateful for the confidence they placed in me.

Finally, I thank all the teachers who encouraged and supported me during my studies.

Dedication

I dedicate this work to my parents:

May they find here the testimony of my deep gratitude and acknowledgment.

And to my brother Abderrahim, and my sisters Rayane and Lina, and to my grandparents, and to my family who provide love and vitality.

And to all my friends who have always encouraged me, I wish them continued success.

Thank you!

Aya KERKAR

ملخص

معظم المساعدين الصوتيين أو الروبوتات الذكية الموجودة في العالم اليوم ليست ذكية بما يكفي لفهم المشاعر. هي مجرد أجهزة تتلقى الأوامر وتنفذها دون أي ذكاء عاطفي. عندما يتحدث الناس مع بعضهم البعض، يفهمون الوضع من خلال نبرة الصوت ويتفاعلون معه، على سبيل المثال، إذا كان شخص ما غاضبا، يحاول الشخص الآخر تهدئته بنبرة صوت ناعمة. هذه التغيرات العاطفية غير ممكنة مع الأجهزة الذكية لأنها تفتقر إلى الذكاء العاطفي. لذا، فإن إضافة المشاعر وجعل الأجهزة تفهم المشاعر سيعزز بشكل كبير قدراتها ويأخذها خطوة أقرب إلى الذكاء البشري. لمعالجة هذا القيد، يقدم نظامنا نهجا جديدا لدمج الذكاء العاطفي في الأجهزة الذكية.

النهج المقترح في هذه الرسالة يتبع سير عمل نموذجي في تعلم الآلة، يشمل تحضير البيانات، وتدريب النماذج، وتقييمها. يستفيد النهج من النماذج المدربة مسبقا ونقل التعلم لاستخراج الميزات من مجموعات بيانات المشاعر، حيث تشمل المكونات الرئيسية استخراج طيف الترددات الميلانية بجانب نموذج التحويل المدرب مسبقا Wev2vec لاستخراج الميزات. تشمل الخطوات الأخرى تقسيم مجموعة البيانات، وضبط نموذج HuBERT المدرب مسبقا لمهام التعرف على المشاعر في الكلام، وتصنيف المشاعر. كما يتيح النظام تحديد جنس المتحدث (ذكر أو أنثى). تم استخدام مجموعات بيانات قياسية مثل RAVDESS و CREMA-D للتدريب والتقييم، وأسفر ذلك عن دقة بنسبة ٨٤.٢٥٪ و ٧١٪ على التوالي.

كلمات مفتاحية: التعرف على المشاعر في الكلام، المحولات، *wav2vec2*، التعلم بالنقل، نموذج *HuBERT* المدرب مسبقا، طيف الترددات الميلانية.

Abstract

Most of the smart devices voice assistants or robots present in the world are not smart enough to understand emotions. They are just like command and follow devices they have no emotional intelligence. When people are talking to each other based on their voice they understand the situation and react to it, for instance, if someone is angry then another person will try to calm him by conveying in a soft tone, these kinds of harmonic changes are not possible with smart devices or voice assistants as they lack emotional intelligence. So adding emotions and making devices understand emotions will significantly enhance their capabilities and take them one step further to human-like intelligence. To address this limitation, our system introduces a novel approach to integrating emotional intelligence into smart devices.

The proposed approach in this thesis follows a typical machine learning workflow, encompassing data preparation, model training, and evaluation. It leverages pre-trained models and transfers learning for feature extraction from emotion datasets, with key components including Mel-frequency spectrogram extraction alongside the Wev2vec pre-trained Transformer model for feature extraction. Other steps involve dataset splitting, fine-tuning the HuBERT pre-trained model for SER, and emotion classification. The system also facilitates speaker gender identification (male or female). Standard datasets RAVDESS and CREMA-D were utilized for training and evaluation, yielding accuracies of 84.25% and 71%, respectively.

Keywords: *speech emotion recognition, transformers, wav2vec2, transfer learning, Hubert pre-trained model, mel spectrograms.*

Résumé

La plupart des appareils intelligents, des assistants vocaux ou des robots présents dans le monde ne sont pas suffisamment intelligents pour comprendre les émotions. Ils se comportent simplement comme des dispositifs de commande et d'exécution et n'ont aucune intelligence émotionnelle. Lorsque les gens se parlent, ils comprennent la situation en fonction de la voix et réagissent en conséquence ; par exemple, si quelqu'un est en colère, l'autre personne essaiera de le calmer en utilisant un ton doux. Ces types de variations harmoniques ne sont pas possibles avec les appareils intelligents ou les assistants vocaux, car ils manquent d'intelligence émotionnelle. Ajouter des émotions et permettre aux appareils de comprendre les émotions améliorera considérablement leurs capacités et les rapprochera un peu plus d'une intelligence semblable à celle des humains.Pour remédier à cette limitation, notre système introduit une nouvelle approche pour intégrer l'intelligence émotionnelle dans les appareils intelligents.

L'approche proposée dans ce mémoire, suit un flux de travail typique en apprentissage automatique, englobant la préparation des données, l'entraînement des modèles et leur évaluation. Elle exploite des modèles pré-entraînés et le transfert d'apprentissage pour l'extraction des caractéristiques des ensembles de données émotionnelles, avec comme composantes clés l'extraction du spectrogramme de fréquences Mel aux côtés du modèle Transformer pré-entraîné Wev2vec pour l'extraction des caractéristiques. D'autres étapes incluent la division des ensembles de données, le peaufinage du modèle pré-entraîné HuBERT pour la REP, et la classification des émotions. Le système facilite également l'identification du genre de l'orateur (homme ou femme). Les ensembles de données standard RAVDESS et CREMA-D ont été utilisés pour l'entraînement et l'évaluation, avec des précisions respectives de 84.25% et 71%.

Mots clés: *reconnaissance des émotions dans la parole, transformers, wav2vec2, transfert d'apprentissage, modèle pré-entraîné HuBERT, spectrogrammes de Mel.*

Contents

List of Figures

List of Tables

List of Abbreviations

HCI: Human-Computer Interactions
AI: Artificial Intelligence
ML: Machine Learning
DL: Deep Learning
TL: Transfer Learning
SER: Speech Emotion Recognition
NLP: Natural Language Processing
EHR: Electronic Health Records
SVM: Support Vector Machine
LR: Logistic Regression
KNN: K-Nearest Neighbor
PCA: Principal Component Analysis
NN: Neural Network
GNNs: Graph Neural Networks
RNN: Recurrent Neural Network
ANN: Artificial Neural Network
CNN: Convolutional Neural Networks
LSTM: Long Short-Term Memory Model
GRU: Gate Recurrent Unit model
DNN: Deep Neural Network
ReLU: Rectified Linear Unit
GELU: Gaussian Error Linear Unit
MSE: Mean Squared Error
CE: Cross-Entropy
FC: Fully Connected
MLP: Multi-Layer Perceptron
CV: Computer Vision
PTMs: Pre-trained Models
FFN: feed-forward network
NLU: Natural Language Understanding

ViT:	Vision Transformer
ASR:	Automatic Speech Recognition
MMSE:	Minimum Mean Square Error
LPC:	Linear Predictor Coefficients
TEO:	Teager Energy Operator
IEMOCAP:	The Interactive Emotional Dyadic Motion Capture
RAVDESS:	Ryerson Audio-Visual Database of Emotional Speech and Song
SAVEE:	Surrey Audio-Visual Expressed Emotion
CREMA-D:	Crowd-Sourced Emotional Multimodal Actors Dataset
EmoDB:	Berlin Database of Emotional Speech
BAVED:	Basic Arabic Vocal Emotions Dataset
EMOVO :	Emotion in Voice
TESS:	Toronto Emotional Speech Set

General introduction

Context

Emotion, encompassing physiological and psychological states, gained systematic attention in the 1990s [Picard, 2000]. Science and technology progress has widely applied emotion recognition in areas like Human-Computer Interactions (HCI) [Nayak et al., 2021], medical health [Colonnello et al., 2019], Internet education [Feng et al., 2020], security monitoring [Fu et al., 2021], intelligent cockpits [Oh et al., 2021], psychological analysis [Sun et al., 2020], and the entertainment industry [Mandryk et al., 2006]. Emotion recognition employs diverse detection methods and sensors, forming human-computer interaction systems[Ogata and Sugano, 1999] or robot systems [Rattanyu et al., 2010]. In medical settings [Hasnul et al., 2021], it aids in patient psychological state detection, supporting treatment, and enhancing medical efficiency. Internet education [Feidakis et al., 2011] utilizes it for assessing students' learning status, improving efficiency through timely reminders. In criminal interrogation [Saste and Jagdale, 2017], it detects lies (authenticity test), and in intelligent cockpits, [Zepf et al., 2020], it enhances driving safety by detecting drowsiness and mental states. Psychoanalysis [Houben et al., 2015] utilizes it for autism analysis, extending to recognizing emotions in individuals unable to express clearly [Bal et al., 2010].

Problem statement

This study aims to improve Speech Emotion Recognition (SER) systems by shifting from traditional machine learning methods, like manual feature engineering with SVMs or GMMs, to Transformer-based models. Traditional approaches, while moderately successful, struggle to capture the subtle patterns in speech that convey emotional nuances due to the complexity of human emotions, including variations in tone, pitch, and timing.

Transformers offer a promising shift for SER due to their ability to capture long-range dependencies and contextual information using self-attention mechanisms. Unlike traditional methods that rely on manual feature engineering, Transformers can autonomously learn relevant features from raw data, potentially enhancing SER system accuracy and robustness. Additionally, leveraging pre-trained Transformer models enables transfer learning, allowing SER systems to efficiently adapt to new tasks with minimal additional data. This transition aims to surpass

the constraints of traditional ML approaches, advancing towards more efficient and adaptive systems for emotional cue recognition in speech.

Objectives

The general objective of this study can be detailed into the following specific sub-objectives:

- Develop a Transformer-based model: for the classification of speech emotions into various categories (e.g., happy, sad, angry, etc.).
- Conduct an experimental study: to determine the optimal hyperparameters and assess the significant impact of these hyperparameters on the model's accuracy.
- Perform a comparative study: to demonstrate the effectiveness of Transformer algorithms in developing efficient speech-emotion recognition systems.

Manuscript organization

This thesis comprises three chapters. Chapter 1 explores AI, ML, DL, Transformers, and transfer learning. Chapter 2 focuses on the state of the art in emotion recognition methods. Finally, Chapter 3 discusses the conception and experimentation process of our system.

- **Chapter 1: Learning in Artificial Intelligence: From Traditional Machine Learning to Transformers and Transfer Learning** This chapter traces the evolution of learning paradigms in AI, from traditional machine learning methods to deep learning, focusing on the Transformer architecture and the role of transfer learning in enhancing model performance for specific tasks.
- **Chapter 2: State of the Art of Speech Emotion Recognition** This chapter covers the theoretical background of emotion recognition and reviews existing methods for speech emotion recognition, starting with traditional techniques involving acoustic feature extraction and classical machine learning, then moving to deep learning approaches like CNNs and RNNs, and culminating in the latest advancements using Transformer-based models, providing a comparative analysis of their effectiveness.
- **Chapter 3: A Transformers-based Speech Emotion Recognition System Using Transformers** This chapter presents a pioneering method for Speech Emotion Recognition (SER) employing Transformers, detailing its implementation and evaluation. It showcases the superior performance of Transformer-based models compared to recent works, underscoring their potential to revolutionize SER systems with improved accuracy and effectiveness.

The system implementation is presented in Appendix A and B.

Chapter 1

Learning in Artificial Intelligence: From Traditional Machine Learning to Transformers and Transfer Learning

1.1 Introduction

Artificial intelligence (AI) and machine learning have transformed many aspects of our daily lives, revolutionizing our ability to process and analyze complex data. The importance of AI lies in its potential to solve problems, enhance efficiency, and drive innovation across various industries. In this chapter, we will explore the different approaches to AI learning. We will start with traditional machine learning methods. Then, we will advance to modern deep learning techniques such as neural networks, transformers, and transfer learning. These techniques are essential for tackling complex tasks and improving real-life applications such as emotion recognition.

The relationship between AI, ML, deep learning, and transformers is shown in the following visual form:

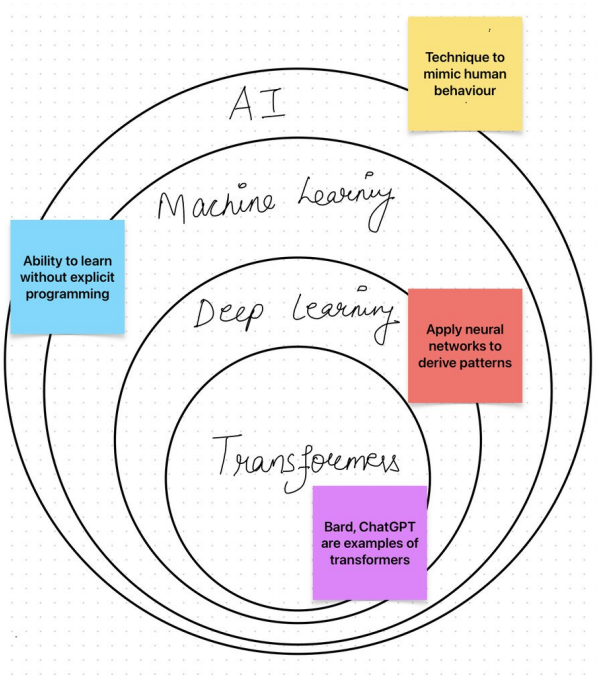

Figure 1.1: AI and ML, Deep learning, Transformers relationships

As depicted in the above diagram:

- The general field is called Artificial Intelligence (AI).
- AI includes machine learning (ML) as a subset.
- Deep Learning is a subset of ML.
- Transformers fall under the domain of deep learning.

1.2 Artificial intelligence

The term Artificial Intelligence (AI) describes a digital computer's or computer-controlled robot to execute tasks typically linked with intelligent entities [Copeland, 2024]. In simpler terms, AI encompasses any combination of software and hardware designed to replicate human behavior and cognitive processes. This broad field encompasses various subfields, such as computer vision, Natural Language Processing (NLP), machine learning, text and speech synthesis, robotics, planning, and expert systems [Mills, 2016]. Figure 1.2 provides a schematic representation illustrating the components that constitute AI.

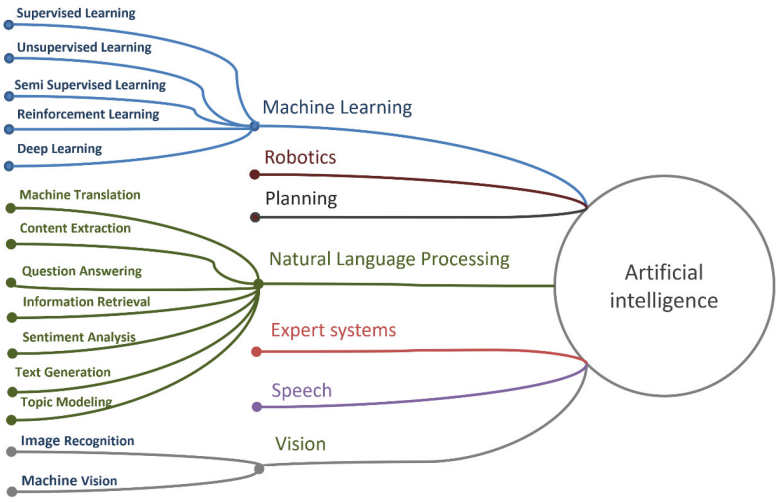

Figure 1.2: Subcategories of artificial intelligence [Mukhamediev et al., 2021]

1.3 Machine learning

Machine Learning (ML), a subset of artificial intelligence, operates as a self-discovery mechanism for data patterns. ML models autonomously learn and adapt without explicit programming, relying on samples rather than predefined rules or limited hypotheses. This approach enhances efficiency, reliability, and cost-effectiveness in computational processes [Bashir et al., 2016]. ML's capability to swiftly and accurately generate models through data analysis is particularly valuable [Coronato et al., 2020]. It proves especially beneficial in handling vast amounts of data, such as health data encompassing demographic information, images, laboratory results, genomic data, medical records, and sensor-derived data [Yousefpoor et al., 2021]. Data sources for ML include network servers, Electronic Health Records (EHR), genomic data, personal computers, smartphones, mobile applications, sensors, and wearable devices [Seaton, 2021].

1.3.1 Types of machine learning techniques

As seen in Figure 1.3, machine learning algorithms can be broadly classified into four types: semi-supervised learning, reinforcement learning, unsupervised learning, and supervised learning [Mohammed et al., 2016]. We go over each kind of learning strategy in brief here, along with how it may be used to address issues in the real world.

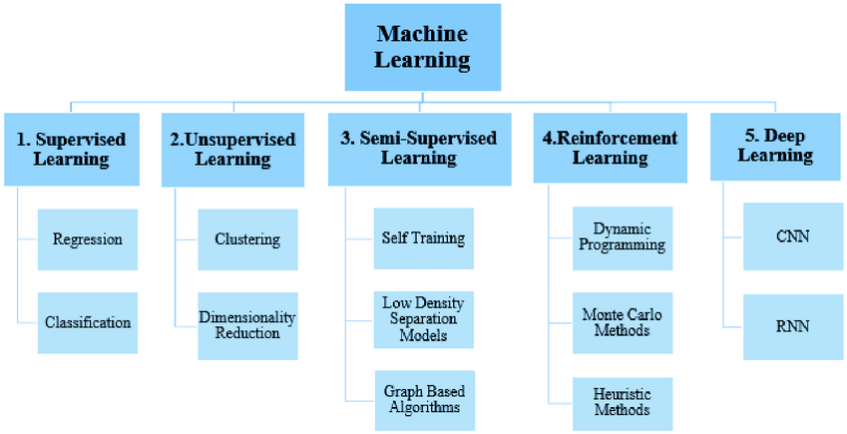

Figure 1.3: Different machine learning types and algorithms [Nassif et al., 2019]

1.3.1.1 Supervised learning

Supervised learning involves the task of learning a function that can map input data to corresponding output based on labeled training examples. In this approach, algorithms require external guidance, learning from a labeled dataset comprising input and output pairs. Typically, the labeled dataset is divided into training and testing subsets, with the training set containing examples with known output variables to be predicted or classified. Supervised machine learning algorithms extract patterns from the training data and apply them to the test data for prediction or classification tasks. Supervised machine learning models are widely used across industries that handle large volumes of organized data. These models excel when data is pre-labeled or categorized, simplifying tasks. They find diverse applications:

- Healthcare benefits from predicting drug interactions, and enhancing patient safety. A study revealed that supervised machine learning accurately forecasts over 90% of harmful drug combinations, potentially reducing adverse events by up to 30%.
- Finance relies on supervised machine learning for precise predictions, like stock prices, and to combat fraud. Major financial institutions, including JPMorgan Chase and Goldman Sachs, heavily invest in this technology.
- Face recognition technology, driven by supervised learning, ensures secure identity validation in various sectors like law enforcement and airport security. Over 95% of face recognition systems use supervised machine learning.
- Voice recognition technology, powered by supervised learning, enhances user interactions with devices. Through datasets and careful analysis, algorithms can understand and respond effectively to spoken commands.
- Meteorologists use supervised machine learning to improve weather forecasting by analyzing past patterns and additional data sources like satellite images. Innovative methods

like Deep Learning Weather Prediction offer precise forecasts for weeks ahead, enhancing traditional methods.

1.3.1.2 Unsupervised learning

This category is termed unsupervised learning, unlike the supervised learning mentioned earlier, where correct answers and a guiding teacher are present. Unsupervised learning algorithms operate independently to uncover and reveal intriguing structures within the data. These algorithms autonomously learn key features from the provided data. Upon the introduction of new data, they utilize the previously acquired features to classify the data's class. Unsupervised learning finds its primary application in clustering and feature reduction. Unsupervised learning finds use in many different fields. Among the noteworthy applications are:

- Customer segmentation: Businesses can customize their marketing campaigns by using unsupervised learning algorithms to group clients according to their purchase patterns.
- Anomaly detection: Unsupervised learning can assist in the detection of fraud, network intrusions, or manufacturing flaws by recognizing anomalous patterns or outliers.
- Picture and text clustering: Unsupervised learning can help with tasks like picture organization, document clustering, and content recommendation by automatically grouping related images or texts.
- Genome analysis: By analyzing genetic data to find patterns and links, unsupervised learning algorithms can provide new insights into genetic research and personalized treatment.
- Social network analysis: Targeted marketing and the identification of online communities can be made possible by the application of unsupervised learning to find prominent individuals or communities within social networks.

[Town,]

1.3.1.3 Semi-supervised learning

Semi-supervised machine learning represents a fusion of both supervised and unsupervised methods in the realm of machine learning. This approach proves fruitful in domains of machine learning and data mining where obtaining labeled data is arduous, and a substantial amount of unlabeled data already exists. Unlike conventional supervised methods that require a labeled dataset for training, semi-supervised learning involves algorithms that can leverage both labeled and unlabeled data. The discussion below delves into some of the semi-supervised learning algorithms. Semi-supervised learning techniques find diverse applications:

- Anomaly detection: These techniques excel in identifying data points significantly different from the rest, utilizing a small amount of labeled data for training and unlabeled data for

anomaly detection. This application is crucial in fraud detection, medical diagnosis, and more.

- Speech analysis: In tasks like speech detection and identification, semi-supervised learning techniques prove beneficial. By initially training the model with labeled data and then leveraging unlabeled data for prediction, these techniques enhance speech analysis. Co-training or self-training methods can be employed for this purpose.
- Internet content classification: With billions of web pages, manually labeling each one is impractical. Search engines simplify this process by employing semi-supervised learning techniques for labeling and ranking internet content, reducing the need for extensive manual work.

1.3.1.4 Reinforcement learning

Reinforcement learning stands as a domain within machine learning that focuses on guiding software agents in making decisions within an environment to maximize cumulative rewards. This paradigm represents one of the three fundamental machine learning approaches, alongside supervised learning and unsupervised learning.

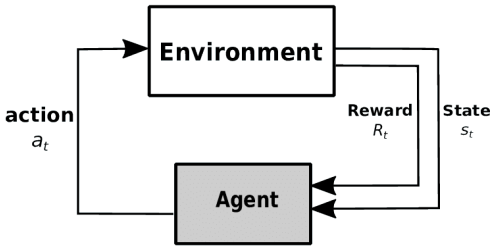

Figure 1.4: Reinforcement learning
[Amiri et al., 2018]

Applications of reinforcement learning

- Autonomous vehicles: In the context of a controller or self-driving car acting as the agent, the environment includes roads, traffic, pedestrians, obstacles, and weather conditions. The agent's actions involve tasks like changing lanes, steering, braking, and accelerating to navigate safely. It receives rewards for efficient and safe travel, but faces penalties for collisions or traffic infractions, emphasizing the need for safe driving practices.
- Robotics: In this setup, the agent functions as a robot controller or an autonomous robot, interacting within a designated physical workspace. Its actions include manipulating objects, navigating obstacles, and performing tasks like grasping items. Rewards are assigned based on the success or failure of these actions, with positive outcomes resulting in rewards and negative outcomes in penalties.

- Automation in industry: In this scenario, the agent takes the form of an automation manager or control system tasked with overseeing operations within a manufacturing environment. Its actions involve optimizing productivity, managing machinery, and fine-tuning production schedules to meet operational goals. Rewards are contingent on the outcomes of these actions, with positive rewards allocated for enhanced efficiency, meeting production targets, and minimizing downtime, while negative rewards are incurred for inefficiencies or disruptions in the production process.

Table 1.1 provides a comparison of the four learning types.

Table 1.1: Comparison of diverse learning types[Rahmani et al., 2021]

Types	**Purpose**	**Dataset**
Supervised learning	Anticipating the classification of the testing set and discerning the connections between inputs and outputs.	Labeled dataset
Unsupervised learning	Recognizing data patterns and categorizing data samples.	Unlabeled dataset
Semi-supervised learning	Predicting the classification or output labels of the testing set.	A dataset containing both labeled and unlabeled data
Reinforcement learning	Determining the optimal course of action by engaging with an environment.	-

1.3.2 Machine learning models

Numerous machine learning algorithms fall under distinct learning categories. These are a few of the most often utilized ones.

1.3.2.1 Models of supervised learning

- Classification

  In machine learning, classification is a supervised learning task where a category is assigned to a new data point based on its features. The features represent the characteristics of the data point utilized for making predictions, and the categories encompass the potential labels assigned to the data point. Among the algorithms used in classification:

Support Vector Machine (SVM): SVM aims to find a max-margin hyperplane in the n-dimensional feature space, providing good results with small training sets due to reliance on a few support vectors. However, sensitivity to noise near the hyperplane is a drawback. While effective for linear problems, SVMs use kernel functions for nonlinear data, mapping it to a new space for separation. Kernel tricks are widely applied in SVMs and other machine-learning algorithms.

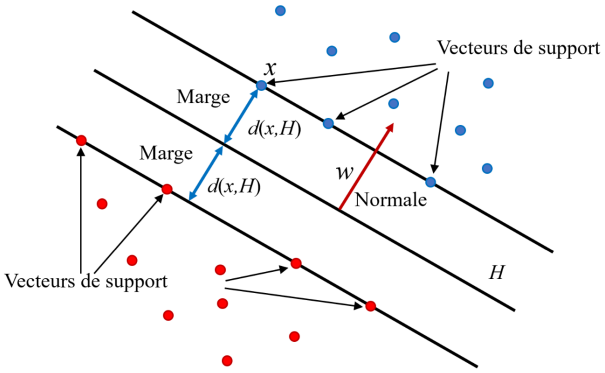

Figure 1.5: Support vector machine
[CNAM,]

Naïve Bayes: Naïve Bayes relies on conditional probability and assumes attribute independence. The classifier calculates conditional probabilities for various classes for each sample, classifying it into the one with the highest probability. The conditional probability is computed using the formula presented in Equation 1.1.

$$P(X = x|Y = c_k) = \prod_{i=1}^{n} P(X^{(i)} = x^{(i)}|Y = c_k) \tag{1.1}$$

The Naïve Bayes algorithm produces the best results when the attribute independence hypothesis is satisfied. But in practice, it is difficult to meet this assumption, which results in less-than-ideal performance, particularly when dealing with attribute-related data.

Logistic Regression (LR): Logistic Regression (LR) is a logarithmic linear model, where the algorithm calculates class probabilities using a parametric logistic distribution, as illustrated in Equation 1.2.

$$P(Y = k|x) = \frac{e^{w_k \cdot x}}{1 + \sum_{k}^{K-1} e^{w_k \cdot x}} \tag{1.2}$$

Here, k = 1,2...K - 1. The sample x is assigned to the class with the highest probability. LR models are straightforward to build and train efficiently. However, LR struggles with nonlinear data, restricting its applicability.

Decision Tree: The decision tree algorithm classifies data using rules, forming an interpretable tree-like model. It automatically excludes irrelevant features during feature selection, tree generation, and pruning. In training, suitable features are chosen individually to create child nodes from the root. It serves as a basic classifier, while advanced methods like random forest and XGBoost consist of multiple decision trees.

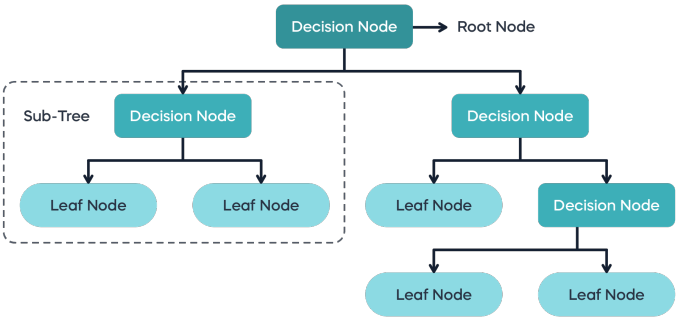

Figure 1.6: Decision tree
[Science, 2024]

Neural Networks: Neural networks are composed of clusters of perceptrons aiming to mimic the neural organization of the human brain. Shallow neural networks, characterized by a sole hidden layer of perceptrons, include examples like collaborative filtering. In this context, the hidden layer of perceptrons undergoes training to capture similarities among entities, facilitating recommendation generation. Platforms such as Netflix, Amazon, and YouTube employ adapted versions of collaborative filtering in their recommendation systems to suggest products aligned with user preferences.

K-Nearest Neighbor (KNN): KNN's fundamental concept follows the manifold hypothesis, where a sample's class likelihood is high if most neighbors share the same class. The classification depends on the top-k nearest neighbors, and the parameter k significantly impacts model performance. A smaller k increases model complexity, elevating the risk of overfitting. In contrast, a larger k simplifies the model, diminishing fitting ability.

- Regression

Regression is a supervised machine-learning technique employed for predicting continuous values. The primary objective of the regression algorithm is to establish a best-fit line or curve that accurately represents the relationship between the data points, and types of regression models:

Linear regression: With the assumption that the dependent and independent variables have a linear relationship, this is the most basic type of regression model.

Logistic regression: used to forecast categorical dependent variables, like the likelihood that a buyer will click on an advertisement.

Polynomial regression: a more complex regression model that considers nonlinear correlations between the independent and dependent variables.

Ridge regression: Implemented to prevent overfitting by incorporating a penalty into the model's coefficients.

Lasso regression: Also employed to prevent overfitting, it achieves this by shrinking the coefficients toward zero.

1.3.2.2 Models of unsupervised learning

- Clustering

Clustering relies on similarity theory, grouping highly similar data into the same clusters and less similar data into different clusters. Unlike classification, clustering is unsupervised learning, requiring no prior knowledge or labeled data. Consequently, data set requirements are relatively low. However, for using clustering algorithms in attack detection, external information reference becomes essential, types of clustering:

K-means clustering: K-means, a standard clustering algorithm, employs K as the number of clusters, and "means" represents attribute means. Utilizing distance as a similarity measure, the algorithm tends to place data objects with shorter distances into the same cluster. While K-means suits linear data, its performance on nonconvex data is suboptimal. The initialization circumstances have an impact on the algorithm. and the parameter K, requiring multiple experiments to determine the appropriate parameter values.

Hierarchical clustering: This algorithm constructs a hierarchy of clusters through the merging or splitting of clusters based on their similarity.

- Association rules:

Association rules in machine learning fall under unsupervised learning, aiming to unveil relationships between variables within a dataset. This technique discovers patterns in data that may not be immediately apparent when examining individual data points. The process involves identifying rules that indicate how frequently two or more items co-occur in a dataset. Association proves to be a potent tool for uncovering latent patterns in data and finding applications across various domains to enhance decision-making processes and efficiency.

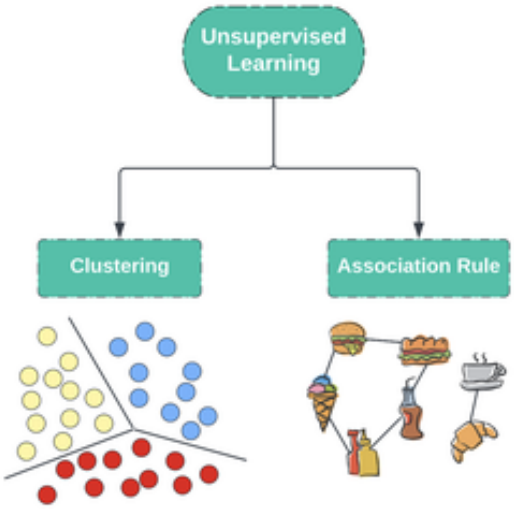

Figure 1.7: Unsupervised learning types – clustering and association rule [Info, 2024]

- Principal Component Analysis (PCA):

  Principal Component Analysis (PCA) is a widely used method in data science for both streamlining model training and visualizing data in lower dimensions. For instance, consider a dataset of images where the goal is to reduce their dimensionality to extract essential features. PCA can efficiently achieve this task, providing a condensed representation of the images that retain critical information while eliminating redundancy. This streamlined approach enhances data analysis and interpretation, facilitating more insightful insights into the underlying patterns within the dataset.

1.3.2.3 Models of reinforcement learning

There are numerous reinforcement learning algorithms categorized into several sub-families. Q-learning is relatively simple and, at the same time, helps understand learning mechanisms common to many other models.

To provide an introductory illustration, a Q-learning algorithm works to solve a basic problem. For example, in the maze game, the objective is to teach the robot to exit the maze as

quickly as possible while randomly placed on one of the white squares. To achieve this, there are three central steps in the learning process:

- Know: Define a Q-value function.
- Reinforce knowledge: Update the Q-function.
- Act: Adopt a strategy of actions (PI).

Consequently, Q-learning is an algorithm for reinforcement learning that seeks to identify the best course of action in light of the existing circumstances. Because the Q-learning function learns behaviors that are against the existing policy, like acting randomly, it is seen as off-policy. The policy is not required. To be more precise, Q-learning aims to discover a policy that maximizes the overall reward. The "Q" in Q-learning stands for quality, denoting how useful a certain activity is in obtaining rewards in the future.[DataScientest, 2024]

1.4 Deep learning

Deep Learning (DL) encompasses methods utilizing deep neural networks, where an artificial neural network with more than one hidden layer is classified as a deep neural network. Conversely, a network with fewer than two hidden layers is termed a shallow neural network.

1.4.1 Neural networks in deep learning

Unlike task-specific algorithms, deep learning is a subset of machine learning that is based on learning data representations. Artificial neural networks imitate the architecture and functions of the brain and are the source of inspiration. To help a computer learn from experience, the method makes use of a hierarchy of concepts in a field. Because this technique gathers computer knowledge automatically, it does not require human input. The hierarchy of concepts makes it easier to dissect intricate ideas into more straightforward ones with multiple layers [Rani et al., 2020]. When there are multiple processing layers, deep learning approaches employ multiple levels of abstraction to learn. This methodology has been particularly useful in the fields of genetics, medicine, and voice and visual object identification. Deep Learning (DL) uses a backpropagation method to identify patterns in complex datasets. To do this, it considers the modifications to the internal parameters required while alternating between levels of representation. Advances in text and speech detection, as well as picture and audio processing, have been made possible by deep convolutional and recurrent nets, respectively. RNN, ANN, and CNN are a few examples of the various implementations of neural networks with minor modifications [LeCun et al., 2015]. Researchers working on challenging deep learning challenges, autonomous cars, and drones favor innovative Neural Network (NN) approaches over machine learning because of their feature engineering and decision limits [Cecchetti et al., 2020]. Any data point can be classified as either positive or negative using the decision boundary technique. Neural

networks are therefore not a good option for deep learning if the data cannot be separated for any reason. However, feature engineering consists of two stages: feature extraction and feature selection. The model building is composed of these two elements. Similar to how neurons are arranged in the human brain, so too are the multi-layer ANNs. Certain coefficients are used to connect each neuron to its neighboring neurons. Information is sent to these connection points during training for them to become familiar with the layout and operation of the network [Mijwil, 2018].

1.4.2 Neural networks structure types

In deep learning, various neural network types, such as ANN [Tabassum et al., 2014], CNN [Chao and Hsieh, 2019], and RNN [Xue et al., 2019], employ different principles to establish rules for diverse applications and serve as the basis for numerous pre-trained models. Additionally, specific neural networks, including radial basis function neural networks, modular neural networks, multilayer perceptron neural networks, and sequence-to-sequence models, leverage their unique structures to excel in specific applications compared to others. DNN [Chen et al., 2019] and Graph Neural Networks (GNNs) cater to graphic data classification issues [Sahu et al., 2015], while LSTM recurrent neural network models prove effective for text classification problems. Examples of applications include using an ANN for simultaneous optimization techniques in modeling theophylline tablet formulations [Hassan et al., 2021] and employing a generative neural network in adjoint electromagnetic simulations [Jiang and Fan, 2020].

1.4.2.1 Artificial neural networks

An Artificial Neural Network (ANN) is made up of several perceptrons or neurons at each layer; this type of network is known as a feed-forward neural network when the input data is sorted forward [Arora et al., 2021]. Three layers make up the fundamental structure of an ANN: the input layer, hidden layers, and output layer. The data is handled by the hidden layers, the output layer outputs the results, and the input layer is responsible for receiving the data. The function of each layer in a neural network is to learn particular decimal weights that will be assigned after the learning process. The ANN technique works well for situations involving text, pictures, and tabular data. An artificial neural network's (ANN) ability to handle nonlinear functions and learn weights to help map any input to any output for any set of data is one of its advantages. The net may learn any complicated relation connected with input and output data by using the activation functions' nonlinear features, which give rise to the concept of a universal approximation. ANNs are often used by academics to tackle difficult relations, such as the cohabitation of WiFi and cellular networks in an unlicensed spectrum [Krizhevsky et al., 2017]. Two further examples are the knowledge-based neural network described in [Rusek et al., 2020], and the feed-forward neural network, also known as a probabilistic neural network (PNN) in [Medsker and Jain, 1999]. This method was applied in [Scarselli et al., 2008] to simulate a solar field in a parabolic trough utilized for direct steam generation. In numerous research projects,

artificial neural networks (ANNs) are utilized as optimizers to address bundling problems. For instance, in [Takayama et al., 2000], an ANN was employed to optimize a rocket's flight path. In [Wu et al., 2016], the design and optimization of microwave circuits was optimized using an ANN. To address wireless system optimization and determine the ideal ANN design, model-aided wireless AI embeds expert knowledge in DNN [Zappone et al., 2019]. Processes for thin film growth are also optimized and controlled using ANN [Alsenwi et al., 2019]. A sampling strategy for the ideal architecture of the ANN model [Kusy and Zajdel, 2014].To create fault tolerance, a feedforward neural network optimization is used [Suganthi et al., 2015]. ANNs are used to investigate nonlinear transformations in conjunction with the Xinjiang model [Na et al., 2016]. To improve the accuracy of bloom forecasting and reduce the expense of aquatic environmental in-situ monitoring, certain improved artificial neural network models for predicting chlorophyll dynamics were developed [Guo et al., 2020]. By improving heat integration, an ANN was used to tackle an issue involving crude oil distillation systems [do Nascimento and de Oliveira, 2017]. ANN used orthogonal arrays to solve the optimization issue and extract anthocyanins from black rice [Rayas-Sánchez, 2004]. The timing traffic light controller's optimization issues are resolved using ANN [Chaffart and Ricardez-Sandoval, 2018]. Additionally, as part of a sustainable optimization of port or coastal defense structures and their conversion for the creation of a predictive model, artificial neural networks (ANNs) are employed as optimizers and applied to Waves Energy Converters (WECs) to anticipate overtopping rates. Figure 1.8 depicts the artificial neural network's architecture. As illustrated in Figure 1.9, each neuron output consists of an activation function equal to the sum of all input weights, whereas the neuron input is the sum of all weights included in the bias. The activation functions are the engine of neural networks, whereas the bias is a constant that modifies the weighted sum of the inputs and the output of the neuron [Zhang et al., 2020, Tian et al., 2017].To obtain the gradients as a neural network with numerous hidden layers, the neural network weight updates are carried out throughout the back-propagation process. During the backward propagation, the gradient may disappear and blow up [Mukherjee et al., 2020, Rusydi et al., 2019].

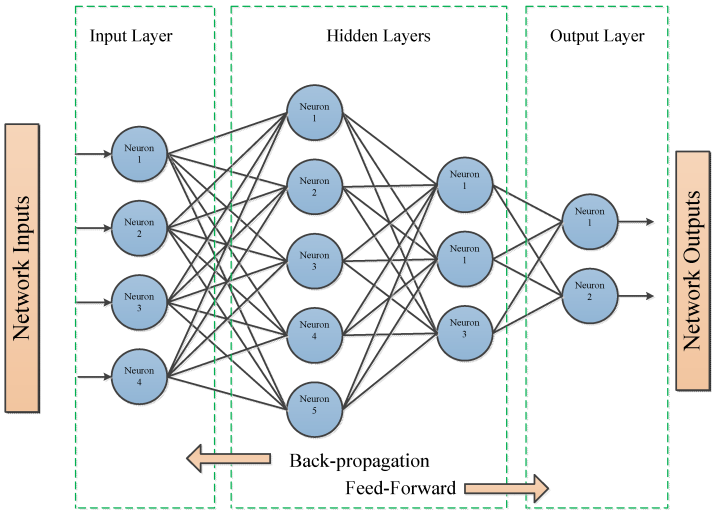

Figure 1.8: Designs of artificial neural networks using backpropagation and feed-forward algorithms

[Abdolrasol et al., 2021]

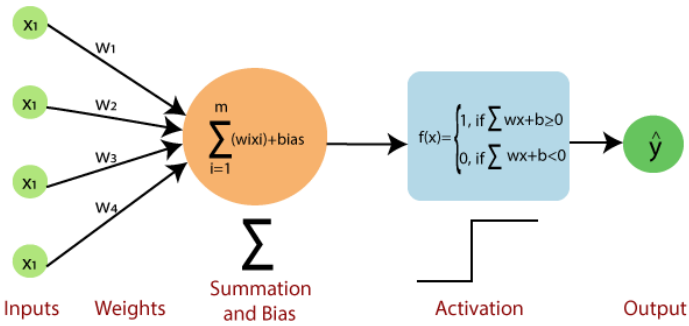

Figure 1.9: Perceptron: a basic neural network model for deep learning

[Ilyurek,]

- Activation functions in neural networks

  Artificial neural networks use specialized activation functions to convert input signals into output signals, which are then fed as input to the subsequent layer in the stack. The output of a layer in an artificial neural network is obtained by applying an activation function to the sum of the products of the inputs and their corresponding weights. This output is then used as the input for the layer below it.

  Why activation functions are needed by neural networks? Neural networks are composed of numerous layers of neurons, or nodes, that are used to classify and predict data when the network receives input data. An input layer, an output layer, and one or

more hidden layers are present. Every layer has nodes, and each node has a weight that is taken into account when information is processed from one layer to the next.

A neural network's output signal would be a simple linear function, or just a degree one polynomial if an activation function is not utilized. Nevertheless, although a linear equation is straightforward and quick to solve, it is limited in its complexity and is incapable of learning and identifying intricate mappings from data. Most of the time, a neural network without an activation function behaves like a weak, linear regression model. It is ideal for a neural network to be able to do more complex tasks than learning and computing a linear function, such as modeling complex data kinds including photos, videos, audio, voice, text, etc.

Because of this, we employ artificial neural network techniques like Deep learning and activation functions to interpret complex, high-dimensional, and nonlinear datasets. These models have multiple hidden layers and complex architectures for knowledge extraction, which is again our ultimate goal [Sharma et al., 2017].

Types of activation functions: The most crucial components of a neural network's architecture are its net inputs, which are processed and converted into an output result known as the unit's activation by applying a scalar-to-scalar transformation function known as the activation function, threshold function, or transfer function. Squashing functions involve permitting a neuron's output in a restricted range and at a constrained amplitude. The output signal's amplitude is condensed into a finite value by a squashing function.

- Logistics function (Sigmoid)

  Because it is non-linear, it is the most often employed activation function. The sigmoid function modifies the values between 0 and 1. The definition of it is

  $$f(x) = \frac{1}{e^{-x}} \tag{1.3}$$

  The sigmoid function is a smooth S-shaped function that is continuously differentiable. The function's derivative is equal to

  $$f(x) = 1 - \text{sigmoid}(x) \tag{1.4}$$

  Additionally, the sigmoid function is not symmetric about zero, meaning that all of the neuronal output values will have the same sign. The sigmoid function can be scaled to address this problem better.[Sharma et al., 2017]

- Hyperbolic Tangent (TanH)

  The function is a hyperbolic tangent. Tanh function is symmetric to about and resembles the sigmoid function. the source. As a result, the outputs from earlier

levels, which serve as input for the subsequent layer, have distinct signs. It has the following definition:

$$f(x) = 2 \cdot \text{sigmoid}(2x) - 1 \tag{1.5}$$

The values of the Tanh function, which is continuous and differentiable, fall between -1 and 1. The gradient of the tanh function is steeper than that of the sigmoid function. Tanh is favored over the sigmoid function because it is zero-centered and features gradients that are not constrained to fluctuate in a certain direction.

– Rectified Linear Unit (ReLU)

The function that most closely resembles its biological equivalent is probably the ReLU function. Many jobs have come to prioritize this function recently, especially those that include computer visions [Oikonomidis et al., 2022]. Similar to the following formula, this function returns x itself if the entry is more than 0 and returns 0 if it is less than 0.The definition of it is:

$$f(x) = \max(0, x) \tag{1.6}$$

– Leaky-ReLU

Leaky ReLU is a modified form of the ReLU function in which the value is defined as extremely tiny for negative values of x rather than zero. component of x that is linear. The following is a mathematical expression for it:

$$f(x) = \begin{cases} 0.01x, & \text{si } x < 0 \\ x, & \text{si } x \geq 0 \end{cases} \tag{1.7}$$

– Gaussian Error Linear Unit(GELU)

More significantly, this function prevents the vanishing gradients problem while also resolving the majority of the preceding activations function problems. In the negative area, it offers a clearly defined gradient and inhibits

Moreover, it works well in transformer models and prevents neurons from dying [Buss, 2023]. The approximate formula for the GELUV is as follows:

$$\text{GELU}(z) = 0.5z \left(1 + \tanh\left[\frac{\sqrt{2}}{\pi}\left(z + 0.044715z^3\right)\right]\right) \tag{1.8}$$

– Softmax

A combination of several sigmoid functions is the softmax function. Since a sigmoid function yields values between 0 and 1, as is known, these can be interpreted as probabilities of data points in a specific class. Softmax function can be used for multiclass classification problems, in contrast to sigmoid functions, which are utilized for binary classification. The function yields the probability for each data point across all individual classes. It is able to be stated as:

$$\text{Softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

The number of neurons in the network's output layer will match the number of classes in the target when we construct a network or model for multiple-class classification.

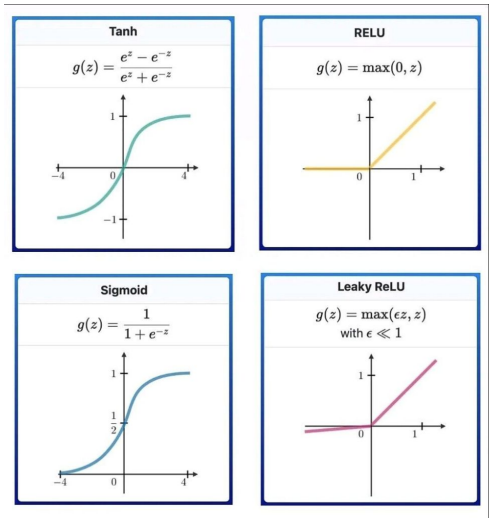

Figure 1.10: Commonly used activation functions

- The cost (loss) functions

  To adjust the weights connecting neurons, neural networks utilize the backpropagation technique based on the cost function, also known as the error function. The cost function represents the overall performance of the global network, typically measured by the average difference between the output and the expected output for a training sample. The cost function depends on network weights, biases, and a specific training sample paired with its expected value. Examples of commonly used cost functions include:

  – Mean Squared Error (MSE)

    Also known as the Quadratic cost function or maximum likelihood, it serves as the default choice for regression problems. The formula to calculate MSE is given by Equation 1.9:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{1.9}$$

    MSE represents the Mean Squared Error, N is the total number of inputs in the training sample, y represents the observed target value, $\hat{y}$ represents the predicted value.

  – Cross-Entropy cost function (CE)

    This function, which is also referred to as negative log-likelihood loss, is frequently employed in machine learning for categorization issues. It calculates the discrepancy between the target variables' actual distribution and the projected probability distribution. Equation 1.10 describes the cross-entropy formula:

$$\text{CE} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{1.10}$$

CE represents the Cross-Entropy, C is the number of classes given in the dataset, y represents the observed target value, $\hat{y}$ represents the predicted value.

1.4.2.2 Convolutional Neural Networks (CNN)

In the realm of DL, CNN is the most well-known and often utilized algorithm [Li et al., 2021, Tomè et al., 2016]. CNN's primary advantage over its predecessors is its ability to identify key information automatically and without human assistance. CNNs are extensively used in many domains, including voice processing, facial recognition, computer vision, and more. Neurons in the brains of humans and animals influence the structure of CNNs, just like in a typical neural network. CNN replicates the intricate cell sequence that makes up a cat's brain's visual cortex. Three major benefits of CNN were noted by Goodfellow [Goodfellow et al., 2020]: parameter sharing, sparse interactions, and comparable representations.CNN utilizes shared weights and local connections to fully utilize 2D input data structures, such as image signals, in contrast to conventional Fully Connected (FC) networks. This method requires a relatively small amount of parameters, which makes it easier and faster to train the network. This is analogous to the visual cortex's cells. It's interesting to note that these cells only see small portions of a scene rather than the full image; in other words, they act like local filters over the input, spatially extracting the available local correlation. The Multi-Layer Perceptron (MLP) [Pedregosa et al., 2011] and a common variant of CNN are comparable in that they have many convolution layers, subsampling (pooling) levels, and FC layers as the final layers. A CNN architecture for image categorization is shown in Figure 1.11. The input (x) for each layer in a CNN model is organized in three dimensions: depth, breadth, and height, or$(m \times m \times r)$, where m is equal to w. The channel number is another name for the term "depth". For example, the depth (r) in an RGB image is three.

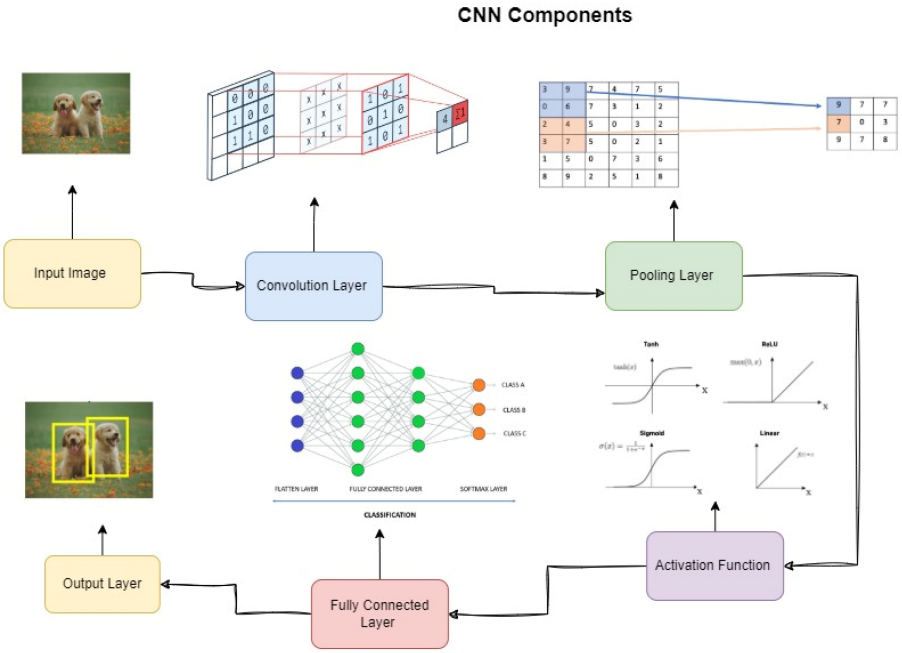

Figure 1.11: The elements of CNN
[Taye, 2023]

Each convolutional layer has several kernels, or filters, denoted by k. These have three dimensions $(n \times n \times q)$, which is similar to the input image; the only differences are that n must be less than m and q must equal or be less than r. Furthermore, the kernels provide the basis for the local connections, which are convolved with input as previously mentioned and have similar properties (bias b^k and weight W^k) to generate k feature maps h^k of size $(m - n + 1)$. Equation 1.11 illustrates how the convolution layer, like NLP, creates a dot product between its input and the weights, but with smaller inputs than the original picture size. Next, we achieve the following by incorporating nonlinearity or an activation function into the convolution layer's output:

$$(h^k = f(W^k \times x + b^k)) \tag{1.11}$$

After that, every feature map in the layers of subsampling is downsampled. As a result, the network's parameters are reduced, hastening the training process and making the overfitting issue easier to solve. For each feature map, a neighboring region of size $(p \times p)$, where p is the kernel size, is subjected to the pooling function (such as maximum or average). After receiving the mid- and low-level data, the FC layers produce the high-level abstraction, which is equivalent to the layers seen in the final stages of a typical neural network. The final layer, such as SoftMax or Support Vector Machines (SVMs), generates the categorization scores [Du and Sun, 2005]. The likelihood of a certain class for a given event is reflected in each score.

1.4.2.3 Recurrent Neural Network (RNN)

A family of neural networks called Recurrent Neural Networks, or RNNs [Venugopal, 2019], is used to analyze sequential input. It is unique in that it can store its past and use it to make predictions (see Figure 1.12). The RNNs use an internal state (which serves as the role) to do this. of a memory) where each output is kept track of. Thus, the present state's (decision's) h_t output is dependent on the previous h_{t-1} output(s).

As a result, the current state's formula is shown as follows:

$$h_t = f(h_{t-1}, x_t) \tag{1.12}$$

Using TanH, the activation function:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{hx}x_t) \tag{1.13}$$

After calculating the current state, we can now compute the output state as follows:

$$y_t = W_{yh}h_t \tag{1.14}$$

Where: x_t is the current input, h_{t-1} is the previous state, W_{hh}, W_{hx} are the weights at the previous hidden state and current input state, respectively, and W_{yh} is the weight at the output state.

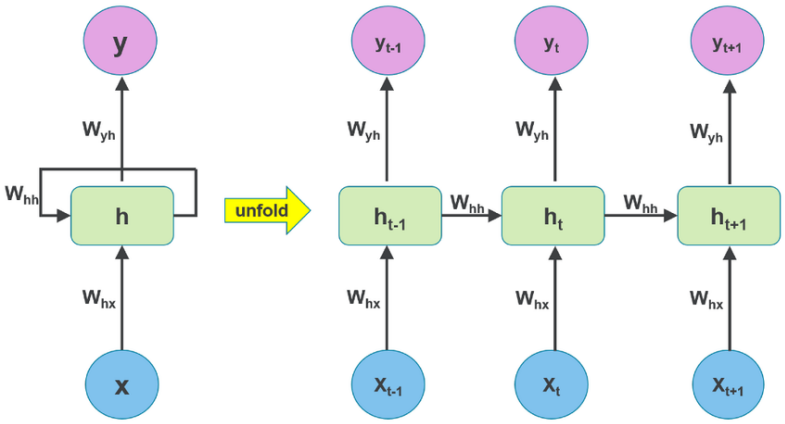

Figure 1.12: The architecture of both the unfolded and simple Recurrent Neural Networks (RNNs)

[Venugopal, 2019]

The LSTM model, introduced by Hochreiter and Schmidhuber in 1997

[Hochreiter and Schmidhuber, 1997], incorporates three gates in each unit: a forget gate, an input gate, and an output gate. The output gate combines short-term and long-term memory to create the present memory state, the forget gate removes out-of-date memory, and the input gate takes fresh data. On the other hand, the GRU was put forth by Chung et al. in 2014 [Chung et al., 2014].

1.4.2.4 Long Short-Term Memory Model(LSTM)

A type of temporal cyclic neural network called a Long Short-Term Memory network (LSTM) was created expressly to solve the long-term dependency problem with a standard RNN (Recurrent Neural Network) [Gers et al., 2000]. In an LSTM network, memory units take the role of the hidden layer neurons of a conventional RNN network. The input, forgetting, and output gates that make up the memory unit's architecture can allow the networks to either retain or erase important data at each time step. An LSTM recurrent network has emerged as one of the top candidate networks in several fields, including speech recognition and language translation, due to its ability to learn temporal correlations. Because these time correlations are dependent on the unpredictable and hard-to-understand behavior of the population, they are frequently observed in power consumption loads. The LSTM network is designed to extract load phases from incoming power consumption profile patterns, store these states in memory, and then forecast based on the acquired information in the case of electrical load forecasting [Kong et al., 2017]. An LSTM cellblock's construction is depicted in Figure 1.13.

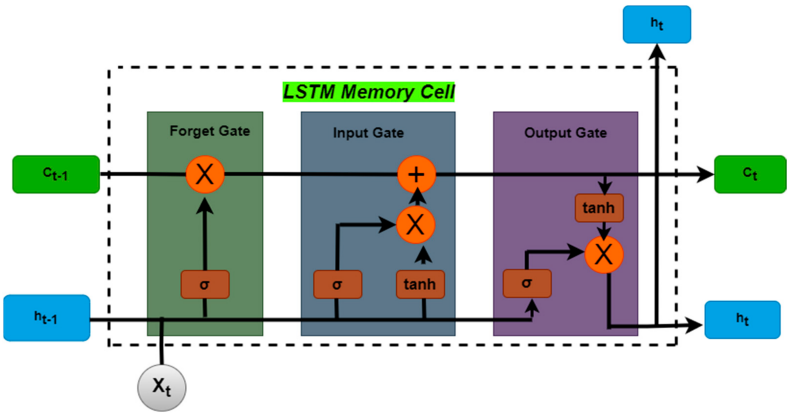

Figure 1.13: A block diagram framework for long-term short-term memory [Abumohsen et al., 2023]

As seen in Figure 1.13, the input gate functions as a filter, excluding any input that is unnecessary for the unit. The forget gate aids in the device's ability to erase any data that was previously kept in memory. This facilitates the unit's ability to concentrate on the fresh data it is getting. The output gate determines whether or not the contents of the memory cell at the output of the LSTM unit should be made public. It has the option to either reveal the contents or not. Because of its sigmoid activation function, this gate can only output a value in the range of 0 to 1. This aids in limiting the gate's output.

1.4.2.5 Gate Recurrent Unit model(GRU)

A gating technique for recurrent neural networks called Gated Recurrent Units (GRUs) was developed in 2014 [Cho et al., 2014]. Since it does not include an output gate, the GRU is comparable to an LSTM with a forget gate but has fewer parameters. In certain tasks such

as voice signal modeling, polyphonic music modeling, and natural language processing, GRU performed better than LSTM [Su and Kuo, 2019]. On smaller and less frequent datasets, GRUs have been shown to perform better [Gruber and Jockisch, 2020]. The schematic and structural representation of GRU, an advancement over the hidden layer of the traditional RNN, is shown in Figure 1.14. An update gate, a reset gate, and a temporary output are the three gates that comprise a GRU. The following are the related symbols:

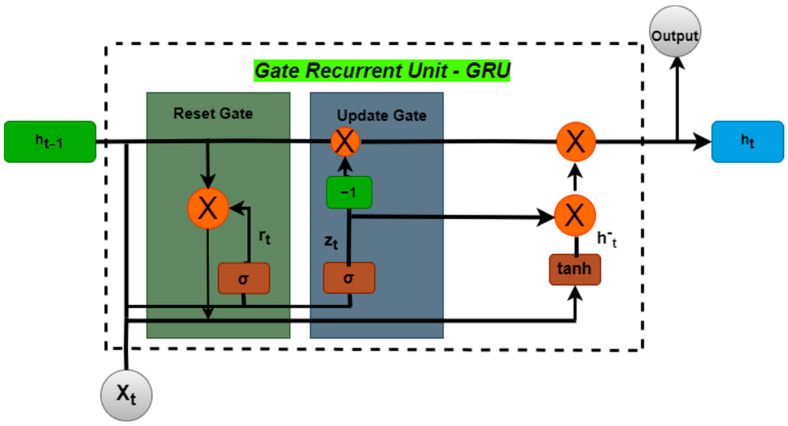

Figure 1.14: Structure of GRU
[Abumohsen et al., 2023]

- The network input at time t is represented by variable xt.
- The information vectors ht and $\overline{h_t}$ represent the temporary output and the hidden layer output at moment t, respectively.
- The gate vectors zt and rt, which represent the output of the update gate and the reset gate at instant t, respectively, are variables.
- (X) and tanh (x) indicate the sigmoid and tanh activation functions, respectively.

1.5 Transformers

Transformer[Vaswani et al., 2017] architecture, has emerged as a dominant deep-learning model with wide-ranging applications across various domains. Initially designed for sequence-to-sequence tasks [Sutskever et al., 2014] like machine translation, Transformer has evolved into a versatile framework utilized in Natural Language Processing (NLP), Computer Vision (CV), speech processing, and beyond. Transformer-based Pre-trained Models (PTMs) have demonstrated exceptional performance across diverse tasks, solidifying the Transformer's status as a go-to architecture in NLP, particularly for PTMs. Beyond language-related applications, Transformer has found utility in CV, audio processing, chemistry, life sciences, and other disciplines. The success of the Transformer has led to the development of numerous variants, often referred to as X-formers, aimed at enhancing the vanilla Transformer from various angles:

- ***Model efficiency:*** Processing long sequences efficiently poses a significant challenge for Transformer due to the computational and memory complexities of the self-attention module. Techniques such as divide-and-conquer strategies and lightweight attention (e.g., recurrent and hierarchical mechanisms) are examples of improvement methods that concentrate on increasing efficiency.

- ***Model generalization:*** The flexibility of the Transformer architecture, coupled with minimal assumptions on the structural bias of input data, makes it challenging to train on small-scale datasets. Strategies for enhancing generalization include introducing structural bias or regularization and pre-training on large-scale unlabeled data.

- ***Model adaptation:*** This line of research aims to tailor the Transformer architecture to specific downstream tasks and applications by adapting its components accordingly, such as fine-tuning or modifying the model's architecture to better suit the target task.

1.5.1 Vanilla transformer

The vanilla transformer[Vaswani et al., 2017], is a sequence-to-sequence architecture comprising an encoder and a decoder, each consisting of a stack of L identical blocks. In each encoder block, there's a multi-head self-attention module and a position-wise Feed-Forward Network (FFN). To construct a deeper model, a residual connection [He et al., 2016] is utilized around each module, followed by layer normalization [Ba et al., 2016]. Compared to the encoder blocks, the decoder blocks have cross-attention modules that connect the position-wise FFNs with the multi-head self-attention modules. Additionally, the self-attention modules in the decoder are adjusted to prevent each position from attending to subsequent positions. The overall architecture of the vanilla transformer is depicted in Figure 1.15. We will outline the essential elements of the vanilla transformer in the next subsection.

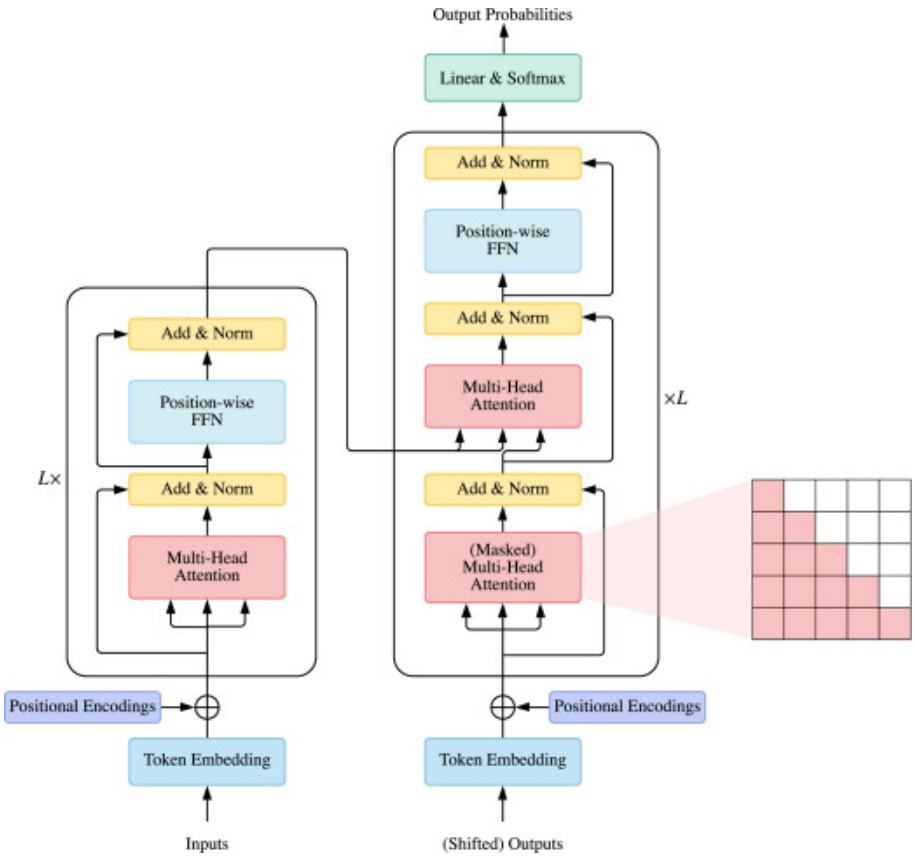

Figure 1.15: An overview of the architecture of the vanilla transformer [Lin et al., 2022]

1.5.1.1 Attention modules

Transformer uses the Query-Key-Value (QKV) model as its attention mechanism. With queries $Q \in \mathbb{R}^{N \times D_k}$, keys $K \in \mathbb{R}^{M \times D_k}$, and values $V \in \mathbb{R}^{M \times D_v}$ as packed matrix representations, the scaled dot-product. Transformer uses the following formula to calculate attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{D_k}}\right)V = AV, \tag{1.15}$$

where N and M represent the lengths of queries and keys (or values) respectively, while D_k and D_v indicate the dimensions of keys (or queries) and values. The attention matrix, denoted as $\mathbf{A}$ and often referred to as the softmax attention, is computed as follows:

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{QK}^\top}{\sqrt{D_k}}\right)$$

Here, softmax is applied in a row-wise manner to the dot-products of queries and keys, divided by $\sqrt{D_k}$ to alleviate the gradient vanishing problem. The Transformer model adopts multi-head attention instead of a single attention function. In this approach, the original queries, keys, and values, each of dimension D_m, are projected into D_k, D_k, and D_v dimensions respectively, using H different sets of learned projections. For each set of projected queries, keys, and values, attention is computed independently according to Equation (1.15). Subsequently, the model

concatenates all the outputs and projects them back to a D_m-dimensional representation, thereby enhancing its ability to capture diverse relationships and patterns in the input data.

$$\text{MultiHeadAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H)\mathbf{W}^O \tag{1.16}$$

$$\text{where} \quad \text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{1.17}$$

In the Transformer framework, there are three distinct types of attention mechanisms utilized based on how queries and key-value pairs are sourced:

Self-attention: Within the Transformer encoder, $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{X}$ in Equation (1.16), where $\mathbf{X}$ denotes the outputs of the preceding layer.

Masked self-attention: In the Transformer decoder, self-attention is restricted so that queries at each position can solely attend to key-value pairs up to and including that position. To enable simultaneous training, a mask function is typically applied to the unnormalized attention matrix $\hat{\mathbf{A}} = \exp\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D_k}}\right)$, where any prohibited positions are suppressed by setting $\hat{\mathbf{A}}_{ij} = -\infty$ if $i < j$. This specific form of self-attention is often known as autoregressive or causal attention (This word appears to have been taken from the causal system, in which the output is dependent on inputs received in the past and present but not in the future).

Cross-attention: Queries emanate from the previous (decoder) layer's outputs, while keys and values are derived from the encoder's outputs.

1.5.1.2 Position-wise FFN

The position-wise Feed-Forward Network 'FFN' (Since the parameters are shared by all positions, two convolution layers with a kernel size of one can also be regarded as the positionwise FFN) is a component within the Transformer architecture. It functions as a fully connected feed-forward module that operates independently on each position in the sequence. The operation of the FFN can be expressed as:

$$\text{FFN}(\mathbf{H}') = \text{ReLU}(\mathbf{H}'\mathbf{W}^1 + \mathbf{b}^1)\mathbf{W}^2 + \mathbf{b}^2 \tag{1.18}$$

Here, $\mathbf{H}'$ represents the outputs from the previous layer, and $\mathbf{W}^1 \in \mathbb{R}^{D_m \times D_f}$, $\mathbf{W}^2 \in \mathbb{R}^{D_f \times D_m}$, $\mathbf{b}^1 \in \mathbb{R}^{D_f}$, and $\mathbf{b}^2 \in \mathbb{R}^{D_m}$ are trainable parameters. Typically, the intermediate dimension D_f of the FFN is set to be larger than D_m. This position-wise FFN allows each position in the sequence to undergo nonlinear transformations independently, facilitating the capture of complex patterns and relationships within the input data.

1.5.1.3 Residual connection and normalization

The transformer uses layer normalization [Ba et al., 2016] after a residual connection [He et al., 2016] around each module to create a deep model. As an illustration, every transformer encoder.

$$H' = \text{LayerNorm}(\text{SelfAttention}(X) + X) \tag{1.19}$$

$$H = \text{LayerNorm}(\text{FFN}(H') + H') \tag{1.20}$$

is one way to write this block, where SelfAttention$(\cdot)$ stands for the self-attention module and LayerNorm$(\cdot)$ for the layer normalization operation.

1.5.1.4 Position encodings

The transformer is unaware of positional information because it does not use convolution or recurrence (particularly for the encoder). To indicate the ordering of tokens, more positional representation is therefore required.

1.5.2 Model usage

In general, there are three ways to use the Transformer architecture:

- The encoder-decoder. There is use of the complete Transformer architecture as described in Section 1.5. Sequence-to-sequence modeling, such as neural machine translation, frequently uses this.
- Just the encoder. There is only one encoder employed, and the input sequence is represented by the encoder's outputs. Natural Language Understanding (NLU) tasks, such as text classification and sequence labeling, frequently use this.
- Just the decoder. All that is utilized is the decoder; the encoder-decoder cross-attention module is eliminated. Usually, sequence generation (like language modeling) uses this.

1.5.3 Model analysis

We examine the two main parts of the Transformer—the self-attention module and the position-wise FFN in Table 1.2 to show the calculation time and parameter needs of the Transformer. Assuming that D is the model's hidden dimension D_m, and assuming that the length of the input sequence is T.the dimension of keys and values is set to D/H, while the intermediate dimension of FFN is set to $4D$. The hidden dimension D predominates in the complexity of position-wise FFN and self-attention when the input sequences are short. The transformer's bottleneck is therefore located in FFN. Nevertheless, the sequence length T progressively takes over the complexity of these modules as the input sequences get longer, at which point self-attention becomes.

Table 1.2: Complexity and parameter counts of position-wise FFN and self-attention

Module	Complexity	#Parameters
Self-attention	$O(T^2 \cdot D)$	$4D^2$
Position-wise FFN	$O(T \cdot D^2)$	$8D^2$

in Table 1.3 highlights The maximum path lengths for various layer types, the minimum number of sequential operations, and the per-layer complexity. The sequence length is represented by T, the representation dimension by D, and the kernel size of convolutions by K

Table 1.3: Maximum Path Lengths, Minimum Sequential Operations, and Per-Layer Complexity for Various Layer Types[Vaswani et al., 2017]

Layer type	Complexity	Sequential operations	Maximum path length
Self-attention	$O(T^2 \cdot D)$	$O(1)$	$O(1)$
Fully connected	$O(T^2 \cdot D^2)$	$O(1)$	$O(1)$
Convolutional	$O(K \cdot T \cdot D^2)$	$O(1)$	$O(\log_K(T))$
Recurrent	$O(T \cdot D^2)$	$O(T)$	$O(T)$

The Transformer bottleneck. Furthermore, the computation of self-attention requires the storage of a $T \times T$ attention distribution matrix, which renders Transformer computation unfeasible in long-sequence applications (such as lengthy text documents and pixel-level modeling of high-resolution images). It will be evident that the objective of raising the long-sequence compatibility of self-attention, as well as the computation and parameter efficiency of position-wise FFN for typical settings, are typically correlated with the Transformer's efficiency.

1.5.4 Comparing transformers to other network types

In this section, we delve into the distinguishing features and performance characteristics of the Transformer architecture relative to other prevalent network types. Our focus is twofold: we first analyze the self-attention mechanism, a cornerstone of the Transformer model, and then examine the inductive biases that differentiate Transformers from recurrent and convolutional networks, as well as Graph Neural Networks (GNNs).

1.5.4.1 Analysis of self-attention

Self-attention, a pivotal component of the Transformer architecture, offers a versatile solution for handling variable-length inputs. It can be conceptualized as a fully connected layer wherein the weights are dynamically determined based on pairwise relations among the inputs. A comparison in Table 1.3 highlights the complexity, sequential operations, and maximum path length of

self-attention against three commonly used layer types. The advantages of self-attention are summarized as follows:

- Equipped with the same maximum path length as fully connected layers, self-attention excels in modeling long-range dependencies. It surpasses fully connected layers in terms of parameter efficiency and adaptability to variable-length inputs.
- Unlike convolutional layers, which necessitate deep network stacking to achieve a global receptive field due to their limited receptive field, self-attention maintains a constant maximum path length. This property enables self-attention to effectively model long-range dependencies without the need for additional layers.
- The consistent number of sequential operations and maximum path length inherent to self-attention render it highly parallelizable and superior in capturing long-range dependencies compared to recurrent layers.

1.5.4.2 In terms of inductive bias

Transformers are frequently contrasted with recurrent and convolutional networks. The inductive biases of translation invariance and locality with common local kernel functions are known to be imposed by convolutional networks. Similar to this, recurrent networks' Markovian structure carries the inductive biases of locality and temporal invariance [Battaglia et al., 2018]. Conversely, the Transformer architecture makes few assumptions regarding the data's structural information. The transformer has a flexible and universal architecture as a result. As a byproduct, Transformer is more likely to overfit small-scale data because of the absence of structural bias.

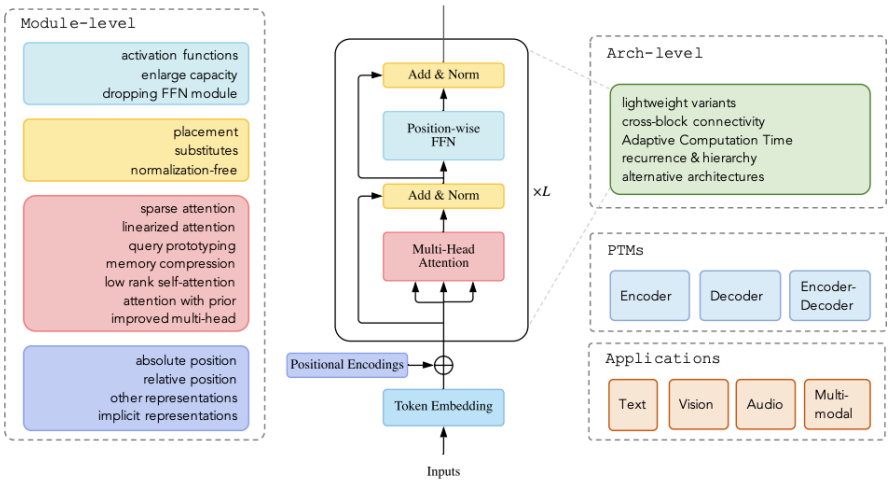

Figure 1.16: Categorization of Transformer variants [Lin et al., 2022]

Graph Neural Networks (GNNs) with message passing are another sort of network that is closely linked [Wu et al., 2020]. Consider a Transformer as a GNN defined over a fully directed

graph (with self-loop) in which every input is represented by a graph node. The primary distinction between Transformer and GNNs is that Transformer passes messages based only on content similarity measures, introducing no prior knowledge about the structure of the incoming data.

1.5.5 Transformers model

In this section, we explore the application of Transformer models across different domains, specifically focusing on computer vision and audio processing. The Transformers has revolutionized these fields by introducing innovative architectures and training methodologies that significantly enhance performance and efficiency.

1.5.5.1 Computer vision

The Vision Transformer (ViT) pioneered convolution-free approaches in computer vision. It employs a conventional Transformer encoder but innovatively treats images by segmenting them into fixed-size patches, akin to tokenizing sentences. Leveraging the efficiency of Transformers, ViT delivered competitive performance compared to CNNs while demanding fewer computational resources. Subsequently, the Swin Transformer emerged, constructing hierarchical feature maps from patches and merging them in deeper layers, resembling CNNs. Attention is confined to local windows, enhancing model learning. Similarly, SegFormer utilizes a Transformer encoder for hierarchical feature mapping but incorporates an MLP decoder for prediction synthesis.

Drawing from BERT's pretraining strategies, models like BeIT and ViTMAE adopt masked image modeling, where patches are randomly masked for pretraining. BeIT predicts visual tokens corresponding to masked patches, while ViTMAE predicts pixels from masked tokens, with 75% of patches masked. Notably, after pretraining, ViTMAE discards the decoder, leaving the encoder ready for downstream tasks.

In decoder-centric models, like ImageGPT, the architecture mirrors text generation models like GPT-2, predicting pixels instead of tokens, suitable for tasks like image generation and potentially image classification post-finetuning. In encoder-decoder frameworks, common in vision models, the encoder extracts crucial image features and passes them to a Transformer decoder. For instance, DETR utilizes a pretrained backbone and a full Transformer encoder-decoder setup for object detection. The encoder learns image representations, combined with object queries in the decoder, predicting bounding box coordinates and class labels for each object query.

1.5.5.2 Audio

The Wav2Vec2 model employs a Transformer encoder to directly learn speech representations from raw audio waveforms. Through pretraining with a contrastive task, it distinguishes true speech representations from false ones. Similarly, Hubert utilizes a Transformer encoder but follows a distinct training approach. It generates target labels through a clustering step, wherein

segments of similar audio are grouped into clusters, serving as hidden units, which are then mapped to embeddings for prediction.

In encoder-decoder architectures, Speech2Text is tailored for Automatic Speech Recognition (ASR) and speech translation. Utilizing log mel-filter bank features extracted from audio waveforms, it is trained to generate transcripts or translations autoregressively. Whisper, another ASR model diverges from conventional approaches by pretraining on a vast dataset of labeled audio transcriptions for zero-shot performance. Notably, a significant portion of this dataset includes non-English languages, enabling Whisper's application in low-resource language scenarios. Structurally akin to Speech2Text, Whisper encodes the audio signal into log-mel spectrograms using the encoder, while the decoder generates transcripts autoregressively based on the encoder's hidden states and previous tokens.

1.6 Transfer learning

A model is trained on a specific domain using labeled data that matches the specified domain in classical supervised learning. Training and testing sets from the same domain or feature space make up the data. For instance, datasets containing annotated photos of several car types are necessary for a machine-learning model to identify distinct car types. When there are insufficient training data available, the supervised learning paradigm collapses. The quantity, caliber, and accuracy of the labeled data determine the model's dependability. For example, an item captured at night cannot be classified by a classification model trained on photographs captured during the day. Since the model has not been exposed to the new domain, its accuracy and performance significantly declined. In specific situations, as when gathering data is costly and risky or there is insufficient data available [Weiss et al., 2016], the model's performance and accuracy deteriorate. When the domain and feature space is the same, other machine learning techniques operate accurately. Nevertheless, the learner models must be retrained to accommodate the new domain when the domain changes. Although it rarely necessitates starting over from scratch or involves the collection of new training data, this retraining procedure is frequently costly in terms of testing and computational effort. With the training that transfer learning offers in these situations, knowledge transfer from one domain to another is feasible. When compared to conventional machine learning methods, transfer learning offers several advantages.

- While traditional methods rely on data, transfer learning uses pre-trained models as a starting point, hence it needs less training data to train the model.
- Transfer learning-trained models can readily generalize to previously unexplored domains. This is so because models for transfer learning are taught to recognize characteristics that can be used in an unknown context or domain.
- Machine learning and deep learning might become more widely available with the help of transfer learning.

- Transfer learning offers an optimal starting point, increased learning accuracy, and quicker training for new domains in contrast to other learning approaches.

As previously indicated, transfer learning appears to offer a more precise model for novel, unknown learning challenges and permits the repurposing of previously developed pre-trained models as a foundation. By avoiding common mistakes, researchers and developers can create innovative, game-changing deep learning and machine learning solutions. The use of costly and time-consuming data collection, cleaning, annotation, and training processes is eliminated via transfer learning. To develop a subject-specific model for removing emotional content from facial datasets, Martina et al. integrated several pre-trained models [Rescigno et al., 2020]. One reason to employ transfer learning is the aforementioned advantages. While the transfer learning technique aims to transfer knowledge from one learning system to another, Figure 1.17 illustrates how standard machine learning systems learn individual tasks from the start. Transfer learning techniques fall into three main categories: inductive, unsupervised, and transductive.

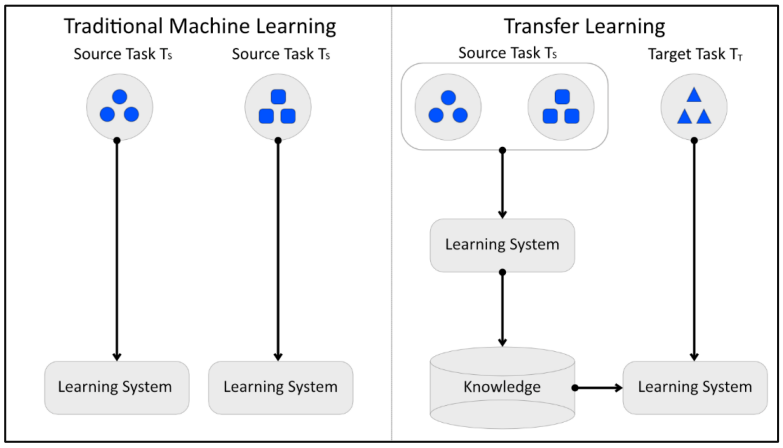

Figure 1.17: Traditional versus transfer learning methods' learning processes [Ranaweera and Mahmoud, 2021]

- Inductive transfer learning: This occurs when there isn't much-labeled data available to be used as target domain training data. In this instance, the creation of an objective model requires some labeled data. This transfer learning approach seeks to enhance the intended function.

- Unsupervised transfer learning: This occurs when the source and target tasks are related but distinct, yet no labeled training data are available from the source and target domains.

- Transductive transfer learning: This refers to the situation in which the source domain has more data accessible while the target domain has no labeled data.

1.7 Conclusion

In this chapter, we explored various AI learning approaches, from traditional methods to advanced deep learning techniques like transfer learning and transformers. These methods have numerous applications in daily life, particularly in speech emotion recognition. In the next chapter, we will examine the state-of-the-art advancements in speech emotion recognition, detailing the latest research and technologies in this exciting field.

Chapter 2

State of the Art of Speech Emotion Recognition

2.1 Introduction

Speech emotion recognition (SER) enhances human-computer interactions by analyzing vocal characteristics to identify emotional states. This chapter explores this field, starting with different emotion models, such as the discrete and dimensional models. We will then discuss the sensors used for emotion recognition, the SER process, relevant datasets, and the applications of this technology. Additionally, we will review the evaluation metrics for SER and highlight recent works in the field.

2.2 Emotion models

The foundation of emotion recognition lies in defining emotion, a concept introduced by Ekman in the 1970s [Ekman, 1971]. Currently, two prevalent emotion models are recognized: The discrete emotion model and the dimensional emotion model.

2.2.1 Discrete emotion model

Darwinian evolution posits that emotions are fundamental, corresponding to discrete and elementary action responses [Darwin and Prodger, 1998]. Emotion, seen as distinct responses or behavioral tendencies, is categorized into limited groups by the discrete emotion model [Ekman et al., 1969]. The discrete emotion model categorizes human emotions into limited types, encompassing happiness, sadness, fear, anger, disgust, surprise, etc. The number of basic emotions varies across theories, ranging from two to eight. These models share common features, viewing emotions as mental and physiological processes triggered by awareness of developmental events. Emotions induce changes in internal and external signals, associated with a fixed set of actions. Ekman identified seven characteristics to distinguish basic emotions, including

autonomous evaluation, specific antecedent events, and rapid onset. Plutchik's wheel model further distinguishes eight basic emotions based on intensity [Plutchik, 2003].

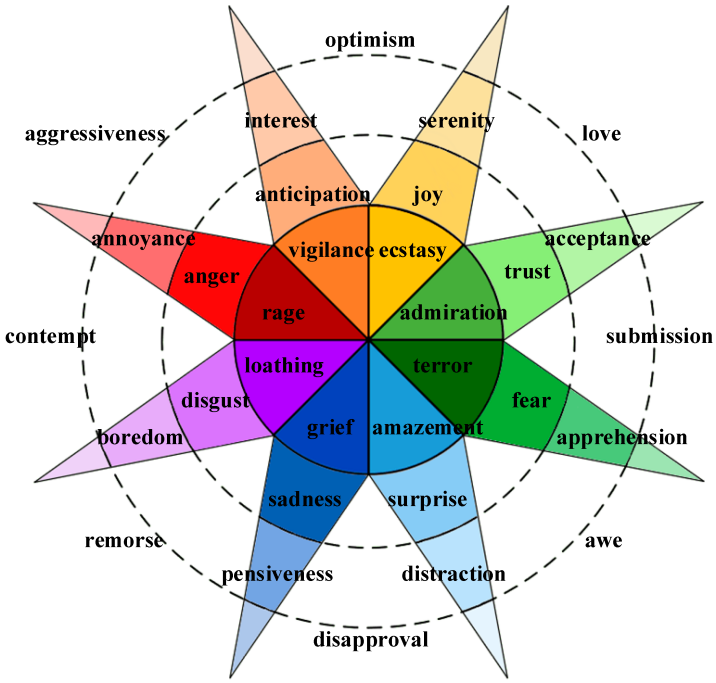

Figure 2.1: Piutchik's wheel model
[Plutchik, 2003]

2.2.2 Dimensional emotion model

Dimensional emotion models conceptualize emotions as vectors within a fundamental dimensional space, simplifying research and measurement of complex emotions. Core emotions are often expressed in two or three dimensions. The two-dimensional arousal-valence model gauges valence, reflecting positive or negative emotion evaluation, and arousal, indicating emotional intensity. However, two-dimensional models struggle to consistently distinguish core emotions with similar arousal and valence, such as anger and fear. To address this, a new dimension is introduced. The renowned three-dimensional pleasure, arousal, and dominance (PAD) model [Bakker et al., 2014], proposed by Mehrabian and Russell through the study of environmental psychology methods [Mehrabian and Russell, 1974] and the feeling-thinking-acting [Bain, 1864]model, as shown in Figure 2.2.

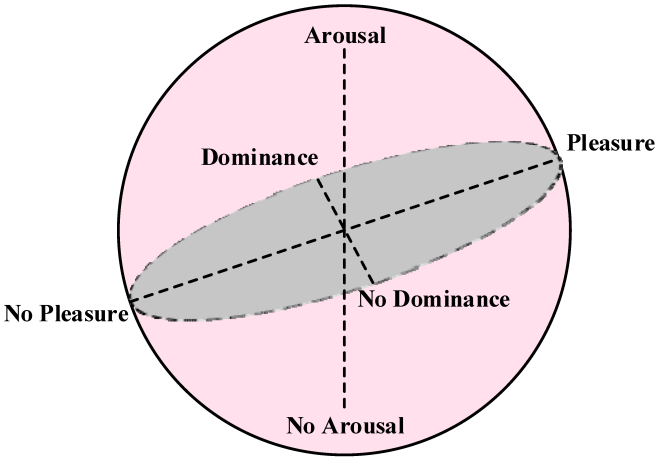

Figure 2.2: PAD 3D emotion model
[Bakker et al., 2014]

Dominance signifies control or position, indicating the submissiveness of a specific emotion. While the dimensional emotion model adeptly identifies core emotions, it may lose nuances for certain complex emotions.

2.3 Sensors for emotion recognition

Sensors for Emotion Recognition utilize various modalities such as physiological, visual, radar, audio, and textual signals. These sensors play a pivotal role in understanding human emotions by capturing and analyzing data from bodily responses, facial expressions, vocal tones, and written communications. In this discussion, we will focus specifically on audio sensors, exploring their significance and applications in deciphering emotional states.

- Visual sensor: A visual sensor [Li and Deng, 2020] is a device capable of capturing visual information, such as images or videos, to detect and analyze facial expressions, body language, and other visual cues associated with emotions. It is commonly used in facial expression recognition systems.

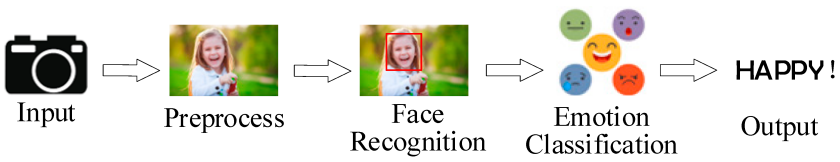

Figure 2.3: Facial expression recognition process
[Cai et al., 2023]

- Radar sensors: Radar sensors [Gouveia et al., 2020] use radio waves to detect and track objects' motion, including subtle movements of the human body. In emotion recognition,

radar sensors can capture physiological responses like chest movements associated with breathing and heartbeats, providing valuable data for analyzing emotional states.

- Physiological sensors: Physiological sensors [Egger et al., 2019] measure various physiological signals, including heart rate, skin conductance, brain activity (via EEG), muscle activity (via EMG), and respiratory rate. These sensors detect changes in the body's physiological responses, which correlate with different emotional states.

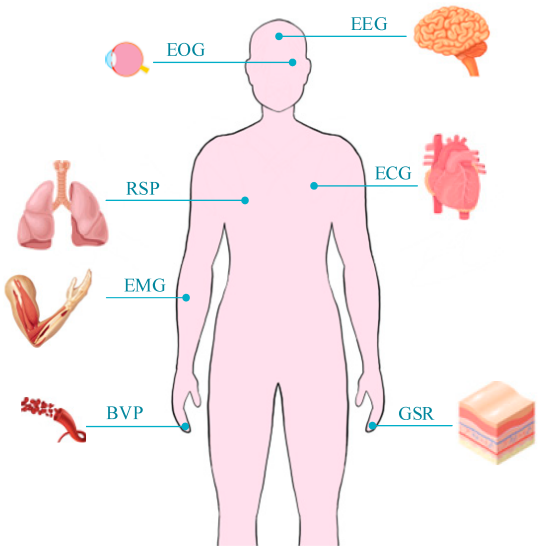

Figure 2.4: Physiological signals detected by other physiological sensors [Cai et al., 2023]

- Textual sensors: Textual sensors [Deng and Ren, 2021] analyze written or textual content, such as emails, chat messages, or social media posts, to extract linguistic features and sentiment analysis. These sensors identify emotional content, sentiment, and mood expressed through written communication.

- Audio sensor: Language, a cornerstone of human culture, facilitates self-expression and communication. Speech recognition [Hinton et al., 2012] has driven the evolution of Speech Emotion Recognition (SER) [Martin et al., 2016]. Recognizing emotions in information is pivotal for effective artificial intelligence engagement in dialogue. Applications of SER include call center dialogues, automatic response systems, autism diagnosis, etc. [Schuller, 2018].

2.4 Speech emotion recognition process

Selecting the appropriate method is pivotal for enhancing the accuracy of emotion recognition [Canal et al., 2022]. Figure 2.5 illustrates the speech emotion recognition process.

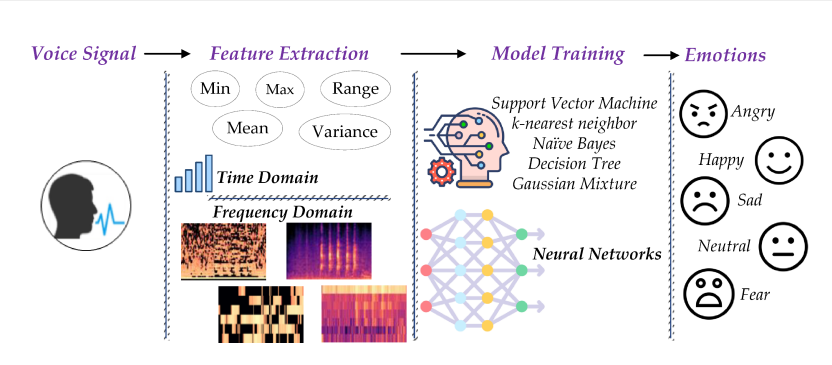

Figure 2.5: The Speech Emotion Recognition (SER) process [Kakuba et al., 2022]

Signal preprocessing aims to enhance signal quality and diminish noise. Feature extraction focuses on identifying distinctive features in different signals, thereby reducing the computational load for classification. Classification involves applying the extracted features to a specific model, ultimately yielding the corresponding emotion through comprehensive analysis.

2.4.1 Signal preprocessing

In emotion recognition from various sensors, signal preprocessing is a critical initial step aimed at mitigating noise impact during the early stages [Kartali et al., 2018].

In audio signal preprocessing

- Silent frame removal is employed to eliminate frames below a predetermined threshold, reducing computational consumption [Nema and Abdul-Kareem, 2018].
- Pre-emphasis compensates for high-frequency components.
- Regularization adjusts the signal to a standard level, diminishing the influence of different environments.
- Windowing prevents signal edge leakage during feature extraction [Beigi, 2011].
- Noise reduction algorithms, such as Minimum Mean Square Error (MMSE), are applied to reduce background noise.

2.4.2 Feature extraction(signal representation)

In feature extraction for speech emotion recognition, there are two types of feature extraction: the first is manual feature extraction, such as MFCCs and spectrograms, and the second is automatic feature extraction, which uses deep learning tools. Here are some examples:

Linear Predictor Coefficients (LPC) [Wong and Sridharan, 2001]: LPC, rooted in a speech production model, utilizes an all-pole filter to characterize vocal tract characteristics, representing the smooth envelope of the speech logarithmic spectrum. Computed directly from windowed speech segments through autocorrelation or covariance methods, LPC efficiently estimates speech parameters. In [Bandela and Kumar, 2018], authors combined TEO and LPC features for T-LPC extraction, achieving precise recognition of stress speech signals with an accuracy of 82.7% (male) and 88% (female) on the Emo-DB dataset. [Idris and Salam, 2015] proposed a spectral coefficient optimization method based on LPC, achieving an 88% accuracy on the Emo-DB dataset, showcasing a 4% improvement through comparative experiments. [Feraru and Zbancioc, 2013] measured emotion recognition accuracy with introduced LPC coefficients, achieving 78% accuracy on the SROL dataset using only LOC coefficients. In [Dey et al., 2020], authors proposed a meta-heuristic feature selection model utilizing LPC features, reaching accuracy rates of 97.31% on SAVEE and 98.46% on Emo-DB datasets.

Teager Energy Operator(TEO)[Bahoura and Rouat, 2001]: TEO, a potent nonlinear energy operator, extracts signal energy based on mechanical and physical considerations, particularly effective for capturing features during stressed speech. It measures speech non-proximity by analyzing signal characteristics in both frequency and time domains. In [Aouani and Ayed, 2020], authors proposed a two-stage emotion recognition system using TEO, enhanced by autoencoders, achieving a 74.07% accuracy on the RML dataset. [Li et al., 2010] introduced the EMD-TEO model, demonstrating robust feature extraction and significant improvement in speech emotion recognition, with an 81.34% accuracy on the EMO-DB dataset. In [Bandela and Kumar, 2017], TEO and MFCC were fused into T-MFCC feature extraction, showcasing superior performance, especially in identifying stressful emotions, with a 93.33% accuracy on the EMO-DB dataset. Additional widely used techniques for extracting features from voice signals are Short-time Coherence (SMC) and Fast Fourier Transform (FFT), Principal Component Analysis (PCA) [You et al., 2006], and linear discriminant analysis (LDA) [Schafer and Rabiner, 1975].

2.4.3 Classification

In classification, we use machine learning and deep learning techniques that we defined in Chapter 1. The classifier can identify various input signals and produce the appropriate emotion category as an output. The accuracy with which emotions are recognized will depend on the classifier's quality.

The trained model step in classification involves: During the training phase, the classifier learns to recognize patterns and features associated with each emotion category from the labeled training data using the machine learning and deep learning techniques defined in Chapter 1. During the classification phase, when a new input signal (e.g., an image, audio signal, or physiological data) is presented to the trained model, it analyzes the features of this signal and

compares them to the patterns learned during training. Based on this analysis, the trained model then assigns the input signal to the corresponding emotion category as output. The accuracy of the classification depends on the quality of the trained model, which is determined by several factors, such as the learning algorithm used, the quantity and quality of the training data, and the model's ability to generalize and capture the important patterns associated with each emotion.

2.5 Applications of speech emotion recognition

Emotion recognition across different modalities has indeed found applications in various domains. Let's explore some notable ones in the context of speech emotion recognition:

- **Healthcare and mental health:** Telemedicine, mental health monitoring, and assessing patient well-being during remote consultations.
- **Customer service:** Sentiment analysis in call centers to identify dissatisfied customers.
- **Human-Computer Interaction (HCI):** Emotion-aware voice assistants and personalized gaming experiences.
- **Education:** Adaptive learning systems and language learning tools.
- **Market research:** Analyzing emotional responses to advertisements.
- **Security and Surveillance:** Detecting potential threats by analyzing voice tones in phone calls or audio recordings.
- **In-car systems:** Information about the driver's mental state can be provided to the car's safety systems to initiate appropriate actions if the driver is detected to be under stress or experiencing negative emotions.
- **Automatic translation systems:** The emotional state of the speaker plays a crucial role in communication between parties, so incorporating emotion recognition can improve translation quality.
- **Mobile communication:** Similar to call centers, emotion recognition can be used to adapt system responses based on the user's emotional state.
- **Diagnostic tool for therapists:** Speech emotion recognition can be used to analyze a patient's emotional state during therapy sessions.

2.6 Evaluation metrics

Each model was assessed using a variety of evaluation measures, including accuracy, precision, recall, and F1-score. A confusion matrix was utilized to identify True-positive (TP), False-positive (FP), True-negative (TN), and False-negative (FN) predictions for each one.

Accuracy: The frequency of sound classes that can be precisely ascertained from the full speech stream is calculated using this metric. The following formula is used to assess whether the results are accurate:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{(TP+TN)_i}{(TP+TN+FP+FN)_i} \tag{2.1}$$

Recall: The following equation is used to check recall to determine how many positive cases the suggested model correctly detects.

$$\text{Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{(TP)_i}{(TP+FN)_i} \tag{2.2}$$

Precision: The following equation is used to verify that the precision approach accurately detected the real utterances.

$$\text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{(TP)_i}{(TP+FP)_i} \tag{2.3}$$

F1-Score: The F1-Score provides a balance between Precision and Recall by taking the harmonic mean of both. This is particularly crucial when there are class imbalances, as seen in equation 4:

$$\text{F1-Score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.4}$$

The performance of models used for detection, classification, and prediction systems is commonly measured using the evaluation matrices used to assess the suggested transformer model. The performance of models used for detection, classification, and prediction systems is frequently measured using the evaluation matrices used to assess the suggested transformer model.

2.7 Datasets for speech emotion recognition

Datasets are crucial for data-driven learning, enhancing model performance and robustness in emotion recognition. Speech emotion recognition datasets are classified into performer-based, induced, and natural datasets based on their acquisition methods. Performer-based datasets involve acted emotions, induced datasets capture emotions in controlled environments, and natural datasets come from real-life conversations.

- **IEMOCAP (The Interactive Emotional Dyadic Motion Capture):** IEMOCAP Multimodal Emotion Recognition With two speakers per session, a total of 302 movies overall from 151 recorded dialogues make up the IEMOCAP dataset. Nine emotions—angry, aroused, fearful, sad, startled, frustrated, glad, disappointed, and neutral—as well as valence, arousal, and dominance are marked for each segment. The dataset was captured throughout five sessions with five speaker pairs.

- **RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song):** The Riley Audio-Visual Database of Emotional Speech and Song there are 7,356 files in the

Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS); the total size is 24.8 GB. Twenty-four professional actors—twelve women and twelve men—vocalize two lexically matched phrases in a neutral North American accent for the database. Expressions of calmness, happiness, sadness, anger, fear, surprise, and disgust are all present in speech, and similar emotions are present in songs. Every expression has two emotional intensity levels (strong and normal), in addition to a neutral expression. Three modalities are offered for all conditions: Video-only (no sound), Audio-Video (720p H.264, AAC 48kHz,.mp4), and Audio-only (16bit, 48kHz.wav). Take note that Actor 18 does not have any song files.

• **CREMA-D (Crowd-Sourced Emotional Multimodal Actors Dataset:)** 7,442 original footage from 91 actors make up the emotional multimodal actor data collection known as CREMA-D. The actors in these clips were 48 men and 43 women, ranging in age from 20 to 74, and representing a range of racial and ethnic backgrounds, including African American, Asian, Caucasian, Hispanic, and Unspecified. Selected from a list of twelve sentences, actors spoke. Six distinct emotions—Anger, Disgust, Fear, Happy, Neutral, and Sad—as well as four distinct emotion levels—Low, Medium, High, and Unspecified—were used to convey the sentences. Based on the combined audiovisual presentation, the video alone, and the audio alone, the participants rated their emotion and emotion levels. Owing to the substantial quantity of evaluations required, this endeavor was crowdsourced, with 2443 individuals in all rating 90 distinct clips—30 audio, 30 visual, and 30 audio-visual. Over seven ratings are present in 95% of the clips.

• **SAVEE (Surrey Audio-Visual Expressed Emotion):** An automatic emotion recognition system requires the recording of the Surrey Audio-Visual Expressed Emotion (SAVEE) dataset. The database includes 480 British English utterances recorded by 4 male actors in 7 different emotional states. The sentences were phonetically balanced for every mood and selected from the standard TIMIT corpus. The data were analyzed, tagged, and recorded in a visual media lab equipped with top-notch audio-visual technology. Ten individuals examined the recordings in auditory, visual, and audio-visual circumstances to assess the quality of performance. Standard characteristics and classifiers were used in the construction of classification systems for the auditory, visual, and audio-visual modalities; speaker-independent identification rates of 61%, 65%, and 84% were attained, respectively.

• **EmoDB (Berlin Database of Emotional Speech):** The German Emotion Database, or EMODB, is openly accessible. The Institute of Communication Science at the Technical University of Berlin, Germany, established the database. Ten expert speakers—five of them male and five of them female—participated in the data collection process. There are 535 utterances in all in the database. There are seven emotions in the EMODB database: Anger, boredom, anxiety, happiness, sadness, disgust, and neutral are the first seven emotions. After being sampled at 48 kHz, the data was down-sampled to 16 kHz.

- **BAVED (Basic Arabic Vocal Emotions Dataset):** BAVED is an Arabic speech dataset containing 1,935 recordings of 7 common words spoken with varying emotion levels by 61 speakers. The words relate to expressing opinions about films. Each word is recorded at low (0), neutral (1), and high positive/negative (2) emotion levels representing tiredness, normal speech, and extreme emotions like joy or anger, respectively. With 45 male and 16 female speakers, the mono 16kHz WAV files provide variety for speech emotion recognition research in Arabic.
- **EMOVO (Emotion in Voice:)** EMOVO is relevant to the Italian language. The voices of up to six actors who spoke 14 lines that simulated the emotions of disgust, fear, rage, joy, surprise, and sadness in addition to the neutral condition made up the database. These feelings are the well-known Big Six, which may be found in the majority of emotional speech-related literature. Professional equipment was used in the laboratories of Fondazione Ugo Bordoni to make the recordings. The study also details a subjective validation test of the corpus, which involved two groups of 24 listeners each, and involved emotion-discrimination of two sentences. The test produced an overall recognition accuracy of 80%, indicating its success. It has been noted that the easiest emotions to identify are anger, sadness, and the neutral state, whereas joy and disgust are the hardest to identify.
- **TESS (Toronto Emotional Speech Set):**Two actresses, ages 26 and 64, each performed a set of 200 target words in the carrier phrase "Say the word ..." The set was recorded depicting each of the seven emotions (anger, disgust, fear, happiness, pleasant surprise, sorrow, and neutral). In total, there are 2800 data points (audio files). The two female actors and their emotions are contained under separate folders in the dataset due to their organizational structure. And all 200 target words audio files can be found within that. The audio file is in the WAV format.

2.8 Recent works on speech emotion recognition

Recent unimodal research on speech emotion identification has concentrated on finding pertinent audio characteristics, like fundamental frequency (pitch), duration, bandwidth, and speech intensity. Speech emotion recognition used techniques like Hidden Markov Models, Gaussian Mixture Models, and Support Vector Machines before deep learning became widely used. These methods involved extensive feature engineering, and changing any feature frequently required rebuilding the system from the ground up. The accuracy of outcomes in controlled situations increased from about 70% to over 90% with the introduction of deep learning in this field [Abbaschian et al., 2021]. The field of speech emotion is followed by the diversity of extracted speech features and the variety of techniques and architectures utilized for acknowledgment. Because recurrent neural networks can simulate temporal aspects, they have shown good performance, particularly bidirectional long short-term memory networks. This method makes use

of both audio elements and spoken words. Self-supervised architectures such as Wav2vec2.0, Whisper, and Herbert have demonstrated encouraging outcomes in voice emotion recognition in recent years. Using the IEMOCAP dataset, Kakouris et al. [Kakouros et al., 2023] fine-tuned WavLM and recorded an accuracy of 75%, the best result yet.

In [Koti et al., 2024], the authors propose an Extreme Machine Learning (EML) approach for SER utilizing the GMM algorithm. EML is a kind of machine learning that achieves great accuracy at a low computing cost by using randomness. An accuracy of 74.33% was obtained when the recommended method was measured using The Berlin Database of Emotional Speech (EMO-DB).

The research by [Xu et al., 2024] provides a new multi-head attention mechanism and GRU network-based speech emotion recognition model. The suggested model achieves 75.04% and 88.93% unweighted accuracy on the IEMOCAP and Emo-DB datasets, respectively.

In [Singh et al., 2021], hierarchical models have been used, achieving SER accuracies of 81.2%, 81.7%, and 74.5% on the RAVDESS, SAVEE, and IEMOCAP datasets, respectively. Additionally, when compared to recently reported methods, these results outperformed them. Thus, the findings indicate that employing a hierarchical deep learning network notably enhances SER compared to standard unimodal and multimodal systems.

In [Ullah et al., 2023], speech emotions using a Transformer encoder for SER and CNN parallelization has been proposed. The effectiveness of the CTENet model for SER is validated by the experimental findings on two widely used benchmark datasets: IEMOCAP and RAVDESS. The authors found that their model outperforms the most advanced models in experiments when it comes to speech emotion recognition, with an overall accuracy of 82.31% and 79.80% for the benchmark datasets.

In [Al-onazi et al., 2022], the researchers provided a unique transformer model based on the fusion of 273 acoustic characteristics. Because Arabic vocal emotions have received relatively little attention in studies, they concentrated on them specifically. The four datasets that this model was used for are BAVED, EMO-DB, SAVEE, and EMOVO. Comparing the experimental results to other methods, it was clear that the suggested model performed admirably. On the BAVED, EMO-DB, SAVEE, and EMOVO datasets, the suggested SER model obtained accuracy values of 95.2%, 93.4%, 85.1%, and 91.7%, respectively. The BAVED dataset yielded the best accuracy, suggesting that the suggested model is a good fit for Arabic vocal emotions.

Other studies, as suggested in [Kwon et al., 2021, Wijayasingha and Stankovic, 2021], also used CNNs and LSTMs to solve the RAVDESS emotion recognition task. These models were fed either preprocessed features or spectrograms, and the results showed accuracies of 80.00% and 81%, respectively.

In [Muppidi and Radfar, 2021] for the RAVDESS, IEMOCAP, and EMO-DB datasets, the model quaternion convolutional neural network obtained an accuracy of 77.87%, 70.46%, and 88.78%, respectively.

The authors of [Kim and Lee, 2023] suggested an approach that uses coordinate information concatenate to improve ViT-based speech emotion identification. By concatenating coordinate

information to the input image, the suggested method preserves pixel location information, which improves CREMA-D accuracy by 82.96% when compared to the state-of-the-art. Consequently, was demonstrated that the coordinate information concatenates suggested in this paper work well for Transformers as well as CNNs.

In [Dal Rì et al., 2023], the researchers integrated a CNN-based model with a Convolutional Attention Block, and they conducted a set of experiments with four English datasets that are commonly used for SER applications: RAVDESS, TESS, CREMA-D, and IEMOCAP. They first tested the proposed pipeline on separate datasets, obtaining mean accuracy of 83%, 100%, 68%, and 63%, respectively. Then, they investigated the generalization capabilities of the extracted features by conducting a thorough cross-validation between common emotional classes that belong to single datasets or combinations of them.

In [DONUK, 2022], the author suggested a technique that uses speech data to increase the accuracy of emotion recognition. This approach uses CNNs to extract additional features from the MFCC coefficient matrices of voice records in the Crema-D dataset. Particle swarm optimization was used to identify the features that are crucial for speech emotion categorization, increasing accuracy by doing so. Furthermore, just 33 attributes instead of 64 were used for every entry. According to the test findings, CNN produced an accuracy of 62.86%, SVM produced an accuracy of 63.93%, and CNN+BPSO+SVM produced an accuracy of 66.01%.

In the study by [Mihalache and Burileanu, 2023], several systems based on deep neural networks (DNNs) with five levels of complexity were proposed. These included systems leveraging transfer learning (TL) for modern image recognition models and ensemble classification techniques to enhance performance. The systems were tested on key SER datasets: EMODB, CREMA-D, and IEMOCAP, for both classification (using full emotion classes and subsets for forensic applications) and regression (using 2D arousal-valence space). The systems achieved state-of-the-art results on EMODB (up to 83% accuracy) and competitive performance on CREMA-D and IEMOCAP (up to 55% and 62% accuracy), especially for negative affective content.

2.9 Conclusion

In this chapter, we explored the state of the art in speech emotion recognition (SER). We discussed various emotion models, including the discrete and dimensional models, and examined the sensors used for capturing emotional data. We outlined the SER process, highlighted the applications and evaluation metrics for SER systems, and discussed key datasets.

Given the superior results achieved by Transformers, our next chapter will focus on our speech emotion recognition system utilizing Transformers, detailing its design, implementation, and performance evaluation.

Chapter 3

A Transformers-based Speech Emotion Recognition System

3.1 Introduction

Speech Emotion Recognition (SER) has emerged as a pivotal area within affective computing, aiming to decipher emotional cues embedded within speech signals. This chapter presents our approach utilizing Transformers for SER, presenting a paradigm shift in how speech data is processed and interpreted. By leveraging the capabilities of Transformers, we aim to enhance the accuracy and robustness of emotion recognition systems, thereby contributing to advancements in human-computer interaction and affective computing research.

3.2 System overview

Our system follows a typical machine learning workflow, involving data preparation, model training, and evaluation. It utilizes pre-trained models and transfer learning to extract features from emotion datasets. The key components include feature extraction using the Wev2vec pre-trained Transformer model, dataset splitting, fine-tuning of the HuBERT pre-trained Transformer for speech emotion recognition task, and emotion classification on input data. The final output is a predicted emotion class, such as neutral, happy, angry, sad, and others. In addition, our system allows speaker gender identification (male or female). The system's performance is evaluated using accuracy, recall, F1-Score, and precision. The general schema of the proposed framework is illustrated in Figure 3.1.

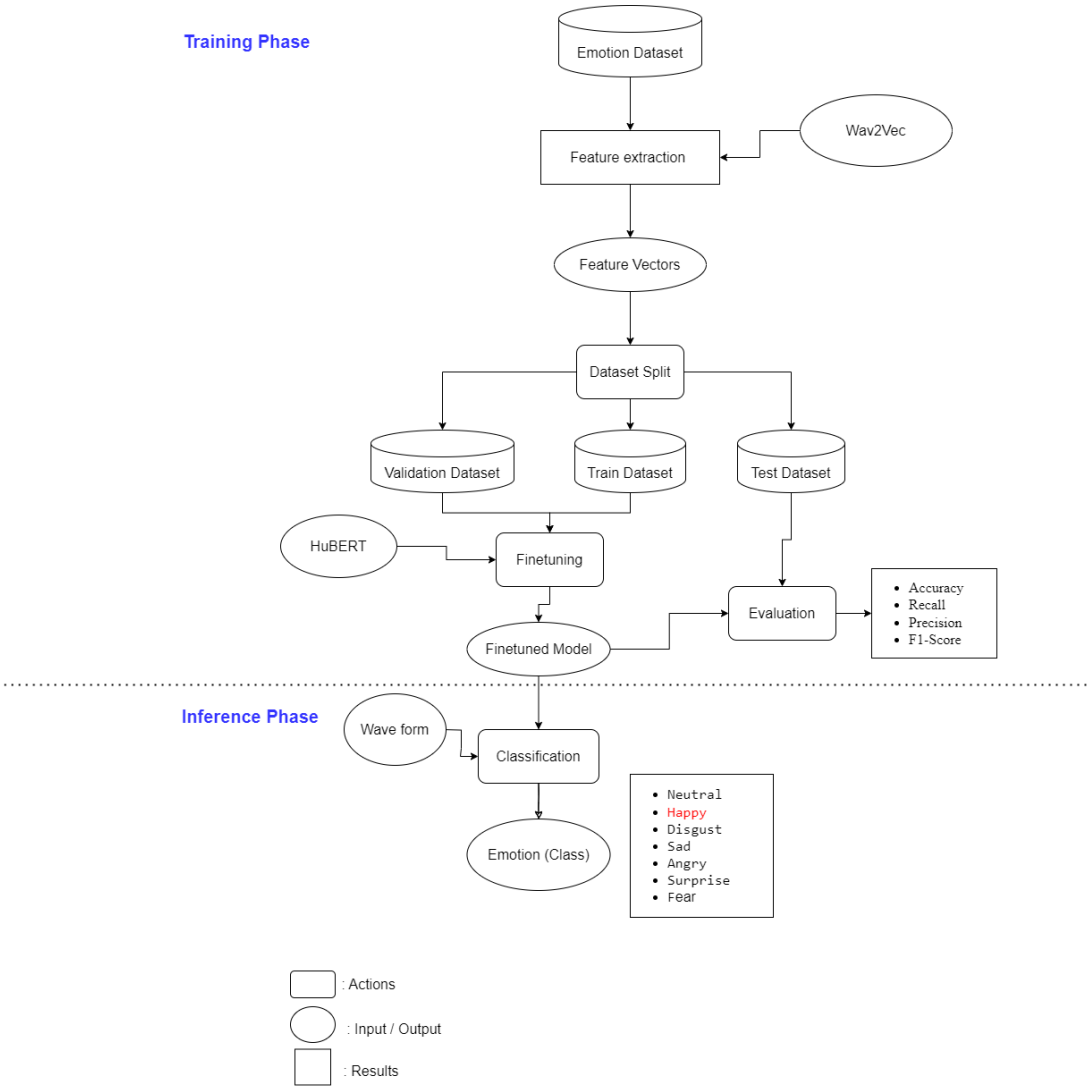

Figure 3.1: Our Proposed Approach

In the following section, we will detail the different components of our system.

3.3 Detailed presentation of our system

This system is designed to recognize emotions from speech. The architecture leverages pre-trained models (Wav2Vec an HuBERT) through a transfer learning approach to accurately classify emotional states. Here's an in-depth look at each component and step involved:

3.3.1 Training phase

The training phase starts with the emotion dataset containing labeled audio samples. We use the Wav2Vec pretrained Transformer model to extract feature vectors from these audio samples. Wav2Vec converts raw waveforms into rich, meaningful features.

These feature vectors are then split into training, validation, and test datasets. The training

dataset is used to train the model, the validation dataset helps tune the model hyper-parameters, and the test dataset evaluates its performance.

For training, we use the HuBERT (Hidden-Unit BERT) pre-trained model. Hubert is fine-tuned with the training dataset, learning to map feature vectors to emotional labels. This fine-tuning process optimizes the model for emotion recognition.

- **Emotion datasets:** The process of training emotion recognition system begins with collecting and curating emotion datasets, which contain labeled samples representing various emotional states. These datasets include speech or audio recordings. The primary purpose of these datasets is to provide the raw data necessary for training, validating, and testing the emotion recognition system, ensuring that the system can accurately learn and identify different emotions from the input data. In our case, we have used the standard datasets RAVDESS and CREMA-D for training and evaluation purposes.

- **Feature extraction:** The feature extraction process involves processing raw data from the emotion datasets using advanced techniques. Specifically, the pre-trained Wav2Vec Transformer is employed for this task. Wav2Vec extracts automaticly meaningful feature vectors, which are numerical representations that capture the relevant information necessary for emotion recognition. These feature vectors are essential as they encode the critical characteristics of the input data, facilitating the subsequent steps in the emotion recognition system.

  - ***Pretrained wav2vec 2.0:*** The wav2vec 2.0 pre-training is conducted in a self-supervised environment and is comparable to the masked language modeling in BERT [Devlin et al., 2018]. The model is trained to replicate the quantized local encoder representations for masked frames at the output of the contextualized encoder, after contiguous time steps from the CNN encoder representations are randomly masked.

    The training goal is demonstrated in Eq. 3.1, where t is the masked time step, Q_t is the union of candidate representations $\tilde{q}$, which includes q_t and $K = 100$ distractors, and κ is the temperature, which is set to 0.1. $\mathrm{Sim}(c_t, q_t)$ is the cosine similarity between the contextualized encoder outputs c_t and the quantized CNN encoder representations q_t. The outputs of the local encoder sampled from masked frames that belong to the same utterance as q_t are the distractors. Next, the comparative loss is determined by adding L_m to all of the masked frames. In order to maximize the utilization of the quantized codebook representations, a diversity loss and an L_2 regularization are applied to the contrastive loss at the conclusion. After a waming up, the learning rate decays linearly, and Adam [Kingma and Ba, 2014] optimizes the pre-training process. In order to enhance ASR performance, wav2vec 2.0 is also adjusted in [Baevski et al., 2020]. A randomly initialized linear projection is appended to the contextual encoder's output for ASR fine-tuning, and the CTC

(Connectionist Temporal Classification [Graves et al., 2006]) loss is reduced. See [Baevski et al., 2020] for further information about wav2vec 2.0's pre-training and fine-tuning.

$$L_m = -\log \frac{\exp(\text{sim}(c_t, q_t)/\kappa)}{\sum_{\tilde{q} \in Q_t} \exp(\text{sim}(c_t, \tilde{q})/\kappa)} \tag{3.1}$$

- **Feature vectors:** Feature vectors are the output of the feature extraction step, consisting of numerical representations that encode the essential characteristics of the input data. Their primary purpose is to serve as the input for training and evaluation steps, enabling the system to learn and recognize different emotional states accurately.

- **Dataset split:** The feature vectors dataset is splited into three subsets: train, validation, and test. The train set is used for model training, allowing the system to learn from the data. The validation set is used for tuning hyperparameters and preventing overfitting, ensuring that the model generalizes well to new data. The test set is used for evaluating the final model's performance, providing an objective measure of how well the system can recognize emotions in unseen data.

  - ***Train set:*** The train set comprises a portion of the feature vectors and is primarily used for model training. During this phase, the model learns from the training data by adjusting its parameters to minimize the predefined loss function. By exposing the model to a diverse range of input samples, the train set allows the system to capture the underlying patterns and relationships within the data, thereby improving its ability to recognize emotions.

  - ***Validation set:*** The validation set is a separate subset of feature vectors used for tuning hyperparameters and preventing overfitting. Hyperparameters are parameters that are not directly learned during training but control the learning process. By evaluating the model's performance on the validation set, adjustments can be made to the hyperparameters to optimize the model's performance and ensure it generalizes well to new, unseen data. This step is crucial for fine-tuning the model and improving its robustness.

  - ***Test set:*** The test set consists of a distinct portion of feature vectors reserved for evaluating the final model's performance. Once the model has been trained and fine-tuned using the train and validation sets, it is evaluated on the test set to provide an objective measure of how well it can recognize emotions in unseen data. The test set serves as a critical benchmark for assessing the model's generalization capability and its ability to perform accurately in real-world scenarios. By analyzing the model's performance on the test set, stakeholders can make informed decisions about deploying the model in practical applications.

- **Transfer learning of HuBERT on speech emotion dataset:** HuBERT (Hidden-unit BERT) is likely a pre-trained model or framework specifically tailored for audio. Its purpose is to provide a strong starting point with pre-learned representations that can be fine-tuned for the specific emotion recognition task, enhancing the model's ability to accurately identify and classify different emotional states from the input data.

  - ***Pretrained HuBERT:*** HuBERT randomly masks CNN-encoded audio features, just like wav2vec 2.0 does. 39-dimensional MFCC characteristics are subjected to a k-means clustering in order to provide labels for the initial iteration of HuBERT pre-training. K-means clustering then operates on the latent characteristics taken out of the HuBERT model that was pre-trained in the previous iteration in order to produce better targets for the following iterations. To forecast cluster labels, a projection layer is placed on top of transformer blocks. When calculating cross-entropy loss over masked timestamps, the following formula is used:

$$L_m(f; X, \{Z^{(k)}\}_k, M) = \sum_{t \in M} \sum_k \log p_f^{(k)}(z_t^{(k)} \mid e^X, t) \tag{3.2}$$

    The set of indices to be masked for a length-T sequence X is indicated by $M \subseteq [T]$, and a corrupted version of X is indicated by $e^X = r(X; M)$, where x_t is substituted with a mask embedding e^x if $t \in M$. The masked prediction model f predicts a distribution $p_f(\cdot \mid e^X; t)$ given an input of e^X over the target indices at each timestep. In the event that an individual clustering model performs poorly, cluster ensembles are used to increase target quality; $Z^{(k)}$ then indicates the target sequences produced by the k-th clustering model. The same learning rate scheduler and optimizer used in wav2vec 2.0 are also used in HuBERT pre-training. The projection layer is eliminated and a randomly initialized softmax layer is added in its stead for ASR fine-tuning. after which the CTC loss is maximized. For additional information about the HuBERT pre-training [Hsu et al., 2021].

- **Finetuning:** Finetuning involves refining the pre-trained HuBERT model on the train dataset using the feature vectors provided by Wav2Vec. This process entails adjusting the model's parameters to better capture patterns in the training data, enhancing its ability to discern subtle emotional cues. As a result of this refinement, the model becomes specialized for recognizing emotions based on the input data, enabling it to provide more accurate and nuanced predictions of emotional states.

- **Finetuned model:** The finetuned model is the result of the finetuning process, embodying a refined version of the pre-trained HuBERT model specifically tailored for the emotion recognition task. It serves the purpose of performing accurate emotion classification on new, unseen data by leveraging its specialized training to discern and categorize emotional states with precision and reliability.

3.3.2 Inference phase

After training, the finetuned HuBERT model is ready to enter the inference phase, where it applies its learned knowledge to predict emotions in new, unseen speech data. The model processes raw speech input through its learned representations to predict emotional states. Inference doesn't involve further parameter adjustments; instead, the model utilizes its trained knowledge to provide accurate assessments of emotions expressed in the speech data.

- **Waveform input:** Waveform input involves utilizing raw audio data, such as audio signals or speech recordings, as input for the finetuned model. This raw audio serves as the primary source for emotion classification tasks, offering real-world samples for the model to analyze and interpret emotional cues accurately. Its purpose lies in enabling the model to directly receive and process raw audio input for emotion classification.

- **Classification:**

  During the classification phase, the finetuned model analyzes the input waveform data and assigns an emotion label to each sample, thereby interpreting the emotional content conveyed in the audio signals or speech recordings. Our system makes possible both emotion and speaker gender identification. In the output, emotions are represented by specific emotion classes. The considered emotion classes are: female_angry, female_disgust, female_fear, female_happy, female_neutral, female_sad, female_surprise, male_angry, male_disgust, male_fear, male_happy, male_neutral, male_sad, male_surprise.

- **Emotion class:** Following the classification phase, the output represents the predicted emotion class or label for the input sample, serving as the final result of the emotion recognition process.

3.3.3 Evaluation

During the evaluation, the emotion recognition system's performance is assessed using the test dataset. Accuracy, recall, precision, and F1-score are the key metrics employed to measure its effectiveness. Accuracy indicates the proportion of correctly predicted emotion labels, while recall gauges the model's ability to identify all relevant instances of emotion. Precision measures the accuracy of the model in identifying true instances of an emotion. Additionally, the F1-score provides a balanced evaluation by considering both precision and recall. We mentioned this in detail in the previous Chapter 2. This comprehensive evaluation framework ensures a thorough assessment, guiding decisions regarding the system's deployment and optimization.

3.4 Experimental results and discussion

In this section, the RAVDESS and CREMA-D datasets, which are used for training and evaluating our model, are first introduced. Then, experiments conducted to optimize model hyper-

parameters are presented. Finally, our obtained results are compared with the state-of-the-art results for the problem of speech emotion recognition.

3.4.1 Datasets used

In this study, we conducted an analysis using two datasets: RAVDESS and CREMA-D. These datasets were introduced in detail in Chapter 2, specifically in Section 2.5.

RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song):

- Contains emotional expressions in both speech and song.
- Includes 24 professional actors vocalizing various emotions such as neutral, happy, sad, angry, fear, surprise, and disgust.

CREMA-D (Crowd-Sourced Emotional Multimodal Actors Dataset):

- Features 91 actors from diverse backgrounds.
- Contains 7,442 clips showcasing a range of emotions including anger, disgust, fear, happiness, sadness, and neutral.

3.4.2 Hyperparameter tuning

For both the RAVDESS and CREMA-D datasets, hyperparameter tuning involves optimizing parameters learning rate, batch size, and Dropout rate to enhance model performance in emotion recognition tasks.

3.4.2.1 For RAVDESS dataset

In this part, we present the experimental results of the Hubert pre-trained model for speech emotion recognition classification. The experiments are conducted to evaluate the performance of the architecture and identify the components and hyperparameters that allow us to obtain the best results. The hyperparameters studied are the number of epochs, the learning rate, the batch size, and the dropout rate. For each experiment, we change the current hyperparameter value and keep the others unchanged.

- **Number of epochs**

  This experiment evaluates the performance of our Hubert-based model across various epochs to determine the best number of epochs. Table 3.1 presents the accuracy on the Train, Validation, and Test datasets for different numbers of epochs.

Table 3.1: Accuracy variation according to he number of epochs

	Epoch 5	Epoch 10	Epoch 15	Epoch 20	Epoch 25	Epoch 30	Epoch 35	Epoch 40
Train	83.92%	99.88%	100	100	100	100	100	100
Validation	56.48%	64.81%	63.89%	65.74%	76.85%	77.78%	76.85%	74.07%
Test	61.11%	74.07%	74.07%	74.07%	77.77%	80.55%	82.40%	68.51%

- Training accuracy: The training accuracy starts at 83.92% at epoch 5 and quickly reaches 99.88% at epoch 10. From epoch 15 onwards, the training accuracy is 100%. This indicates that the model learns the training data very quickly and achieves perfect accuracy by epoch 15.
- Validation accuracy: The validation accuracy starts at 56.48% at epoch 5 and fluctuates but generally increases to 77.78% at epoch 30. There is a slight decrease to 76.85% at epoch 35 and further to 74.07% at epoch 40. This trend suggests that the model is learning to generalize better up to epoch 30, after which there might be some signs of overfitting as the validation accuracy starts to decrease.
- Test accuracy: The test accuracy starts at 61.11% at epoch 5 and improves to a peak of 82.40% at epoch 35. The accuracy then drops to 68.51% at epoch 40. The highest test accuracy is observed at epoch 35, indicating that this epoch is likely the optimal point where the model has learned sufficiently without overfitting.
- Best epoch number based on test dataset: Epoch 35 is the best epoch based on the test dataset, with the highest accuracy of 82.40%. This suggests that the model performs best on unseen test data at epoch 35, balancing between learning enough from the training data and not overfitting.

- **Learning rate**

Table 3.2 presents accuracy on the training, validation, and test datasets obtained using different learning rates. Each row corresponds to a specific learning rate, and the columns represent the accuracy achieved on each dataset. The validation accuracy shows variability with different learning rates, without a clear trend of improvement or degradation. However, the test accuracy demonstrates significant variability, ranging from 75.92% to 84.25%, indicating the sensitivity of model performance to the choice of learning rate. Notably, a learning rate of 0.003 achieves the highest test accuracy of 84.25%, suggesting its effectiveness in generalizing to unseen data. This sensitivity underscores the importance of careful selection of the learning rate during model training to optimize performance and generalization. Additionally, monitoring validation accuracy can provide insights into potential overfitting or underfitting during training.

Table 3.2: Accuracy variation according to learning rates

Learning Rate (Lr)	**Train**	**Validation**	**Test**
0.001	100%	76.85%	82.40%
0.002	99.88%	75%	82.40%
0.003	100%	72.22%	84.25%
0.004	99.77%	71.30%	82.40%
0.005	99.88%	71.30%	75.92%

- **Batch-size**

  Table 3.3 illustrates the impact of different batch sizes on the accuracy of a trained model across the training, validation, and test datasets. Batch size refers to the number of samples processed by the model in each training iteration. This analysis aims to investigate how varying batch sizes influence the model's performance and generalization ability.

Table 3.3: Accuracy variation according to batch size

Batch Size	Train	Validation	Test
4	100%	72.22%	84.25%
8	100%	66.07%	82.14%
16	100%	66.43%	73.21%
32	99.88%	67.97%	61.45%

  Training accuracy remains consistently high (99.88% to 100%), indicating the model's proficiency in learning the training data irrespective of batch size. However, validation accuracy varies slightly (66.07% to 72.22%), suggesting a minor impact of batch size on validation performance. Test accuracy fluctuates notably (61.45% to 84.25%), demonstrating the significant influence of batch size on model generalization. The highest test accuracy (84.25%) is observed with a batch size of 4, indicating the potential benefits of smaller batch sizes. Nevertheless, this trend is inconsistent across all batch sizes, emphasizing the need for careful batch size selection to balance training efficiency and model performance on unseen data.

- **Dropout rate**

  Table 3.4 displays accuracy on the train, validation, and test datasets across different dropout rates. Dropout is a regularization technique used in neural networks to prevent overfitting. This experiment explores the impact of varying dropout rates on the model's performance and generalization.

Table 3.4: Accuracy variation according to dropout rates

Dropout	Train	Validation	Test
0.1	100%	80.56%	75.92%
0.3	100%	72.22%	84.25%
0.5	100%	78.70%	74.07%
0.7	99.53%	68.52%	75.0%
0.9	100%	80.56%	84.25%

  Training accuracy remains consistently high (99.53% to 100%), indicating robust learning regardless of dropout rate. Validation accuracy shows a minor variation (68.52% to

80.56%) with dropout changes. Test accuracy fluctuates notably (74.07% to 84.25%), suggesting dropout rate significantly impacts model generalization. The highest test accuracy (84.25%) is observed at a dropout rate of 0.3, indicating its potential efficacy. However, this trend varies, underlining the importance of careful dropout rate selection to balance training efficiency and model performance on unseen data.

We selected the hyper-parameters shown in Table 3.5 to be used in the construction of our model based on the trials we conducted.

Table 3.5: Hyper-parameters used on RAVDESS dataset

Hyper-parameter	**Value**
Batch-size	4
Epochs	35
Dropout rate	0.3
Learning rate (Lr)	0.003
Optimizer	SGD
Loss function	CrossEntropyLoss

3.4.2.2 For CREMA-D dataset

We studied the CREMA-D dataset for our approach following the same steps as the previous RAVDESS dataset. The following results were obtained for this dataset, along with the hyper-parameter specifications. We also evaluated the model performance to ensure the effectiveness and robustness of our approach, as shown in Table 3.6, which details the hyper-parameters used in the approach.
The hyper-parameters used in the approach.

Table 3.6: Hyper-parameters used for CREMA-D dataset

Hyper-parameter	**Value**
Batch-size	4
Epochs	30
Dropout rate	0.3
Learning rate (Lr)	0.001
Optimizer	SGD
Loss function	CrossEntropyLoss

3.4.3 Model performance evaluation

Evaluating model performance is crucial to understanding how well it generalizes to new data and identifying areas for improvement

3.4.3.1 On RAVDESS dataset

We tested the model on 10% of the entire dataset after it had been trained using the hyper-parameters shown in the above table. Figure 3.2, and Table 3.7 displays the obtained results.

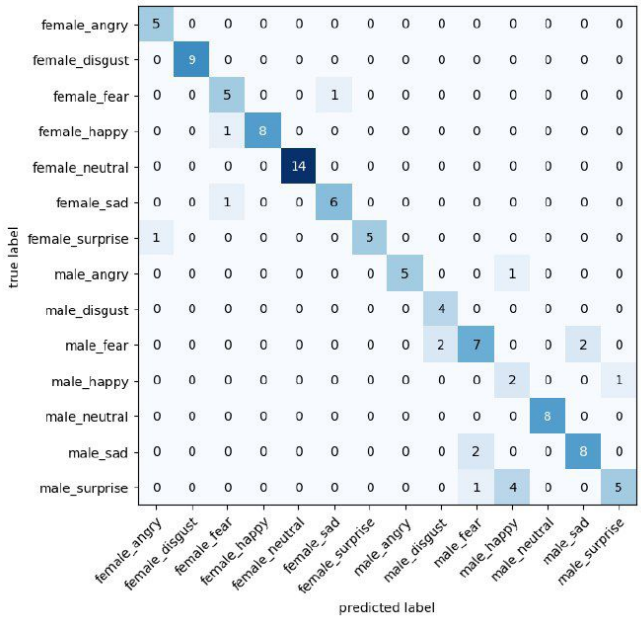

Figure 3.2: Model test results "Confusion matrix"

The table below summarizes the performance metrics of our system, which classifies speech into various emotional categories. The metrics include precision, recall, and F1-score for each emotion category, as well as overall accuracy, macro average, and weighted average. These metrics are essential for evaluating the effectiveness of the SER system in accurately detecting and classifying different emotional states from speech data.

Table 3.7: Performance Metrics for Speech Emotion Recognition

Emotion	Precision	Recall	F1-Score	Support
female_angry	0.83	1.00	0.91	5
female_disgust	1.00	1.00	1.00	9
female_fear	0.71	0.83	0.77	6
female_happy	1.00	0.89	0.94	9
female_neutral	1.00	1.00	1.00	14
female_sad	0.86	1.00	0.92	6
female_surprise	1.00	0.83	0.91	6
male_angry	1.00	0.67	0.80	3
male_disgust	0.67	1.00	0.80	9
male_fear	1.00	0.64	0.78	11
male_happy	0.29	1.00	0.44	3
male_neutral	1.00	1.00	1.00	10
male_sad	0.80	0.80	0.80	10
male_surprise	0.83	0.50	0.62	10
accuracy	-	-	0.84	108
macro avg	0.84	0.85	0.85	108
weighted avg	0.87	0.84	0.85	108

3.4.3.2 On CREMA-D dataset

After training the model with the hyper-parameters displayed in Table 3.6, we tested it on 10% of the whole dataset. Figures 3.3 and Table 3.8 show the results that were achieved.

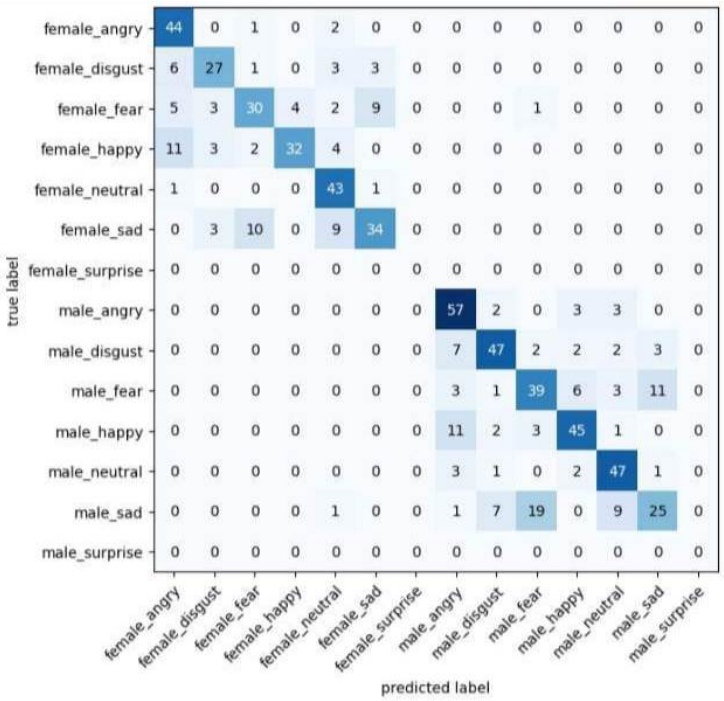

Figure 3.3: Model test results "Confusion matrix" for CREMA-D

The confusion matrix provides a detailed evaluation of the Speech Emotion Recognition (SER) system's performance. Rows represent true labels and columns represent predicted labels, with high diagonal values indicating correct predictions.

The system shows strong performance in recognizing emotions like female anger (44 out of 49) and male anger (57 out of 69). It also accurately classifies female neutral (43 out of 44) and male neutral (47 out of 50).

However, there are areas of confusion, particularly for emotions with overlapping acoustic features. Female happiness (32 out of 52) is often confused with female fear and male fear, while male happiness (45 out of 97) shows significant misclassification. Female sad (34 out of 56) is frequently mistaken for female fear. Both female and male disgust show moderate performance but are confused with other emotions, indicating challenges in finer distinctions.

The table below summarizes the performance metrics of a Speech Emotion Recognition (SER) system, detailing precision, recall, and F1-score for each emotion category, along with overall averages.

Table 3.8: Performance Metrics for Speech Emotion Recognition

Emotion	Precision	Recall	F1-Score	Support
female_angry	0.66	0.94	0.77	47
female_disgust	0.75	0.68	0.71	40
female_fear	0.68	0.50	0.58	42
female_happy	0.89	0.69	0.78	52
female_neutral	0.67	0.62	0.64	54
female_sad	0.72	0.70	0.71	56
female_surprise	0.79	0.68	0.73	47
male_angry	0.70	0.88	0.78	65
male_disgust	0.78	0.62	0.69	63
male_fear	0.61	0.62	0.61	55
male_happy	0.61	0.58	0.59	47
male_neutral	0.72	0.87	0.79	54
male_sad	0.70	0.70	0.70	62
male_surprise	0.62	0.40	0.49	49
micro avg	0.71	0.71	0.71	663
macro avg	0.61	0.61	0.60	663
weighted avg	0.71	0.71	0.70	663

3.5 Comparison of results

Our system's accuracy was compared to some recent works conducted using RAVDESS data to assess its performance in other studies that are currently available. The comparison's findings are presented in the bar chart.

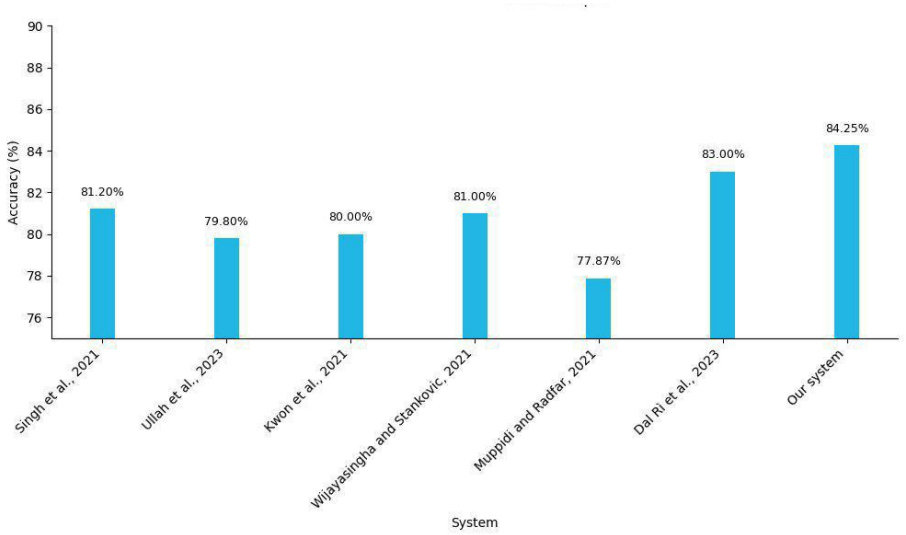

Figure 3.4: Performance comparison of recent rorks on the RAVDESS dataset

The comparison of our proposed system with recent works on the RAVDESS dataset reveals notable performance variations. Singh et al. (2021) employed hierarchical models, reaching 81.2%, while Ullah et al. (2023) used a Transformer encoder with CNN parallelization, attaining 79.8%. Traditional CNN and LSTM methods by Kwon et al. (2021) and Wojpayingcha and Srisamorw (2021) achieved accuracies of 80.0% and 81.0%, respectively. Muppidi and Rudrar (2021) utilized a Quaternion CNN with a 77.87% accuracy. Dai Rì et al. (2023) integrated a CNN-based model with Convolutional Attention Blocks, resulting in 83%. Our system stands out with an 84.25% accuracy, highlighting its effectiveness in speech emotion recognition.

We now evaluate our system's performance in other existing studies by comparing its accuracy on the CREMA-D dataset. The results of the comparison are shown in the bar chart.

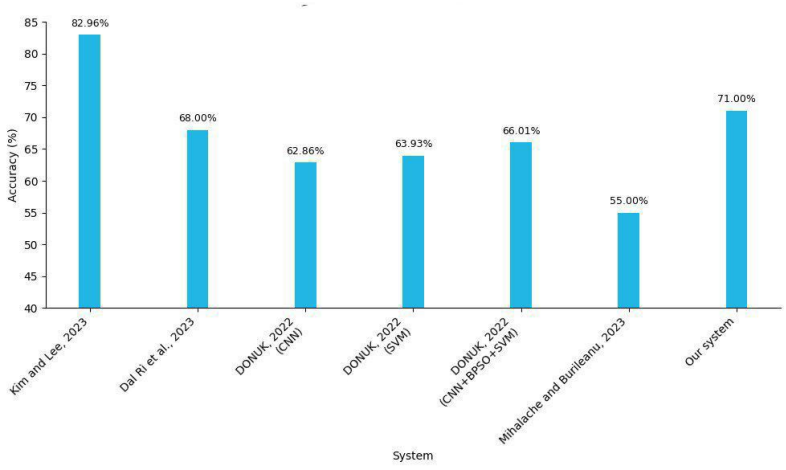

Figure 3.5: Performance comparison of recent rorks on the CREMA-D dataset

The performance of various methods on the CREMA-D dataset has been compared in re-

cent works, shedding light on the effectiveness of different approaches in emotion recognition. Among these studies, the work by Kim and Lee (2023) stands out with Transformers achieving an impressive accuracy of 82.96%. In contrast, Dal Rì et al. (2023) utilized CNNs and attained a slightly lower accuracy of 68%. DONUK (2022) explored CNNs, SVMs, and a combined approach, with accuracies ranging from 62.86% to 66.01%. Additionally, Mihalache and Burileanu (2023) employed DNNs, yielding an accuracy of 55%. In comparison, our system achieved a commendable accuracy of 71%. While not the highest among the listed methods, this performance underscores the competitive nature of our system in emotion recognition tasks on the CREMA-D dataset.

3.6 Conclusion

This chapter has presented a comprehensive framework for SER employing Transformers. We introduced the architecture and methodology of our approach, detailing both the training and inference phases. Through extensive experimentation, we evaluated the performance of our model using datasets such as RAVDESS and CREMA-D, highlighting the effectiveness of hyperparameter tuning for optimizing results. Furthermore, we compared our findings with existing approaches, showcasing the competitive edge of our system in accurately discerning emotions from speech data. Overall, our work demonstrates the potential of Transformers in advancing the field of SER, paving the way for more sophisticated and nuanced emotion recognition systems with wide-ranging applications in human-computer interaction, healthcare, and beyond.

General conclusion

Conclusion

In this thesis, the critical need for emotional intelligence in smart devices, such as voice assistants and robots, is emphasized. Current devices lack the ability to understand and respond to human emotions, limiting their interaction effectiveness. Addressing this gap, the proposed approach leverages machine learning workflows and pre-trained models to enhance speech emotion recognition (SER) systems. By utilizing Mel-frequency spectrograms and advanced models Wev2Vec and HuBERT, the system achieves significant improvements in emotion classification accuracy. The experimental results demonstrate that the Transformer-based approach yields accuracies of 84.25% on the RAVDESS dataset and 71% on the CREMA-D dataset, underscoring the potential of these models to revolutionize SER systems.

Perspectives

Future research should explore several avenues to build on the findings of this thesis:

- Expanding the datasets used for training and evaluation to include more diverse and naturalistic speech samples, which could improve the generalization capabilities of SER models.
- Integrating multimodal emotion recognition, combining speech with facial expressions and physiological signals, to provide a more holistic understanding of human emotions.
- Deploying these enhanced SER systems in real-time applications, such as customer service, healthcare, and education, to evaluate their performance in real-world scenarios.
- Exploring the ethical implications and ensuring privacy and data security in emotion recognition systems for broader societal acceptance and implementation.
- Advancing the interpretability of Transformer-based models to better understand their prediction processes and build trust in their decision-making.

Bibliography

[Abbaschian et al., 2021] Abbaschian, B. J., Sierra-Sosa, D., and Elmaghraby, A. (2021). Deep learning techniques for speech emotion recognition, from databases to models. *Sensors*, 21(4):1249.

[Abdolrasol et al., 2021] Abdolrasol, M. G., Hussain, S. S., Ustun, T. S., Sarker, M. R., Hannan, M. A., Mohamed, R., Ali, J. A., Mekhilef, S., and Milad, A. (2021). Artificial neural networks based optimization techniques: A review. *Electronics*, 10(21):2689.

[Abumohsen et al., 2023] Abumohsen, M., Owda, A. Y., and Owda, M. (2023). Electrical load forecasting using lstm, gru, and rnn algorithms. *Energies*, 16(5):2283.

[Al-onazi et al., 2022] Al-onazi, B. B., Nauman, M. A., Jahangir, R., Malik, M. M., Alkhammash, E. H., and Elshewey, A. M. (2022). Transformer-based multilingual speech emotion recognition using data augmentation and feature fusion. *Applied Sciences*, 12(18):9188.

[Alluhaidan et al., 2023] Alluhaidan, A. S., Saidani, O., Jahangir, R., Nauman, M. A., and Neffati, O. S. (2023). Speech emotion recognition through hybrid features and convolutional neural network. *Applied Sciences*, 13(8):4750.

[Alsenwi et al., 2019] Alsenwi, M., Yaqoob, I., Pandey, S. R., Tun, Y. K., Bairagi, A. K., Kim, L.-w., and Hong, C. S. (2019). Towards coexistence of cellular and wifi networks in unlicensed spectrum: A neural networks based approach. *IEEE Access*, 7:110023–110034.

[Amiri et al., 2018] Amiri, R., Mehrpouyan, H., Fridman, L., Mallik, R., Nallanathan, A., and Matolak, D. (2018). A machine learning approach for power allocation in hetnets considering qos. *IEEE Access*, PP.

[Ang et al., 2002] Ang, J., Dhillon, R., Krupski, A., Shriberg, E., and Stolcke, A. (2002). Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In *Proceedings of INTERSPEECH*, pages 2037–2040, Denver, CO, USA.

[Aouani and Ayed, 2020] Aouani, H. and Ayed, Y. (2020). Speech emotion recognition with deep learning. *Procedia Computer Science*, 176:251–260.

[Arora et al., 2021] Arora, V., Mahla, S. K., Leekha, R. S., Dhir, A., Lee, K., and Ko, H. (2021). Intervention of artificial neural network with an improved activation function to predict the

performance and emission characteristics of a biogas powered dual fuel engine. *Electronics*, 10(5):584.

[Ba et al., 2016] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

[Baevski et al., 2020] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

[Bahoura and Rouat, 2001] Bahoura, M. and Rouat, J. (2001). Wavelet speech enhancement based on the teager energy operator. *IEEE Signal Processing Letters*, 8:10–12.

[Bain, 1864] Bain, A. (1864). *The Senses and the Intellect*. Longman, Green, Longman, Roberts, and Green, London, UK.

[Bakker et al., 2014] Bakker, I., Van Der Voordt, T., Vink, P., and De Boon, J. (2014). Pleasure, arousal, dominance: Mehrabian and russell revisited. *Current Psychology*, 33:405–421.

[Bal et al., 2010] Bal, E., Harden, E., Lamb, D., Van Hecke, A., Denver, J., and Porges, S. (2010). Emotion recognition in children with autism spectrum disorders: Relations to eye gaze and autonomic state. *Journal of Autism and Developmental Disorders*, 40:358–370.

[Bandela and Kumar, 2017] Bandela, S. and Kumar, T. (2017). Stressed speech emotion recognition using feature fusion of teager energy operator and mfcc. In *Proceedings of the 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–5.

[Bandela and Kumar, 2018] Bandela, S. and Kumar, T. (2018). Emotion recognition of stressed speech using teager energy and linear prediction features. In *Proceedings of the 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, pages 422–425. IEEE.

[Bashir et al., 2016] Bashir, S., Qamar, U., Khan, F. H., and Naseem, L. (2016). Hmv: A medical decision support framework using multi-layer classifiers for disease prediction. *Journal of Computational Science*, 13:10–25.

[Battaglia et al., 2018] Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

[Beigi, 2011] Beigi, H. (2011). *Fundamentals of Speaker Recognition*. Springer Science & Business Media, Berlin, Germany.

[Burmania and Busso, 2017] Burmania, A. and Busso, C. (2017). A stepwise analysis of aggregated crowdsourced labels describing multimodal emotional behaviors. In *Proceedings of INTERSPEECH*, pages 152–156, Stockholm, Sweden.

[Buss, 2023] Buss, J. (2023). Activation function gelu in bert. *OpenGenus IQ*.

[Cai et al., 2023] Cai, Y., Li, X., and Li, J. (2023). Emotion recognition using different sensors, emotion models, methods and datasets: A comprehensive review. *Sensors*, 23(5):2455.

[Canal et al., 2022] Canal, F., Müller, T., Matias, J., Scotton, G., de Sa Junior, A., Pozzebon, E., and Sobieranski, A. (2022). A survey on facial emotion recognition techniques: A state-of-the-art literature review. *Information Sciences*, 582:593–617.

[Cecchetti et al., 2020] Cecchetti, R., de Paulis, F., Olivieri, C., Orlandi, A., and Buecker, M. (2020). Effective pcb decoupling optimization by combining an iterative genetic algorithm and machine learning. *Electronics*, 9(8):1243.

[Chaffart and Ricardez-Sandoval, 2018] Chaffart, D. and Ricardez-Sandoval, L. A. (2018). Optimization and control of a thin film growth process: A hybrid first principles/artificial neural network based multiscale modelling approach. *Computers & Chemical Engineering*, 119:465–479.

[Chao and Hsieh, 2019] Chao, K.-H. and Hsieh, C.-C. (2019). Photovoltaic module array global maximum power tracking combined with artificial bee colony and particle swarm optimization algorithm. *Electronics*, 8(6):603.

[Chen et al., 2019] Chen, D., Zou, F., Lu, R., and Li, S. (2019). Backtracking search optimization algorithm based on knowledge learning. *Information Sciences*, 473:202–226.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

[Chung et al., 2014] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[CNAM,] CNAM, C. Machine learning course material. https://cedric.cnam.fr/vertigo/Cours/ml2/. Accessed: 2024-05-17.

[Cohen, 1984] Cohen, R. (1984). A computational theory of the function of clue words in argument understanding. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, pages 251–258, Stanford University, Stanford, CA, USA.

[Colonnello et al., 2019] Colonnello, V., Mattarozzi, K., and Russo, P. (2019). Emotion recognition in medical students: Effects of facial appearance and care schema activation. *Medical Education*, 53:195–205.

[Copeland, 2024] Copeland, B. J. (2024). Artificial intelligence. *Encyclopedia Britannica*.

[Coronato et al., 2020] Coronato, A., Naeem, M., De Pietro, G., and Paragliola, G. (2020). Reinforcement learning for intelligent healthcare applications: A survey. *Artificial Intelligence in Medicine*, 109:101964.

[Dal Rì et al., 2023] Dal Rì, F. A., Ciardi, F. C., and Conci, N. (2023). Speech emotion recognition and deep learning: an extensive validation using convolutional neural networks. *IEEE Access*.

[Darwin and Prodger, 1998] Darwin, C. and Prodger, P. (1998). *The Expression of the Emotions in Man and Animals*. Oxford University Press, Oxford, UK.

[DataScientest, 2024] DataScientest (2024). Q-learning: Le machine learning avec apprentissage par renforcement. Accessed: 2024-05-17.

[Dellaert et al., 1996] Dellaert, F., Polzin, T., and Waibel, A. (1996). Recognizing emotion in speech. In *Proceedings of the Fourth International Conference on Spoken Language Processing (ICSLP'96)*, pages 1970–1973, Philadelphia, PA, USA.

[Deng et al., 2017] Deng, J., Frühholz, S., Zhang, Z., and Schuller, B. (2017). Recognizing emotions from whispered speech based on acoustic feature transfer learning. *IEEE Access*, 5:5235–5246.

[Deng and Ren, 2021] Deng, J. and Ren, F. (2021). A survey of textual emotion recognition and its challenges. *IEEE Transactions on Affective Computing*, 14(1):49–67.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Dey et al., 2020] Dey, A., Chattopadhyay, S., Singh, P., Ahmadian, A., Ferrara, M., and Sarkar, R. (2020). A hybrid meta-heuristic feature selection method using golden ratio and equilibrium optimization algorithms for speech emotion recognition. *IEEE Access*, 8:200953–200970.

[do Nascimento and de Oliveira, 2017] do Nascimento, E. O. and de Oliveira, L. N. (2017). Numerical optimization of flight trajectory for rockets via artificial neural networks. *IEEE Latin America Transactions*, 15(8):1556–1565.

[DONUK, 2022] DONUK, K. (2022). Crema-d: Improving accuracy with bpso-based feature selection for emotion recognition using speech. *Journal of Soft Computing and Artificial Intelligence*, 3(2):51–57.

[Du and Sun, 2005] Du, C.-J. and Sun, D.-W. (2005). Pizza sauce spread classification using colour vision and support vector machines. *Journal of Food Engineering*, 66(2):137–145.

[Egger et al., 2019] Egger, M., Ley, M., and Hanke, S. (2019). Emotion recognition from physiological signal analysis: A review. *Electronic Notes in Theoretical Computer Science*, 343:35–55.

[Ekman, 1971] Ekman, P. (1971). Universals and cultural differences in facial expressions of emotion. In *Nebraska Symposium on Motivation*, Lincoln, NE, USA. University of Nebraska Press.

[Ekman et al., 1969] Ekman, P., Sorenson, E., and Friesen, W. (1969). Pan-cultural elements in facial displays of emotion. *Science*, 164:86–88.

[Feidakis et al., 2011] Feidakis, M., Daradoumis, T., and Caballé, S. (2011). Emotion measurement in intelligent tutoring systems: What, when and how to measure. In *Proceedings of the 2011 Third International Conference on Intelligent Networking and Collaborative Systems*, pages 807–812, Fukuoka, Japan.

[Feng et al., 2020] Feng, X., Wei, Y., Pan, X., Qiu, L., and Ma, Y. (2020). Academic emotion classification and recognition method for large-scale online learning environment-based on a-cnn and lstm-att deep learning pipeline method. *International Journal of Environmental Research and Public Health*, 17:1941.

[Feraru and Zbancioc, 2013] Feraru, S. and Zbancioc, M. (2013). Emotion recognition in romanian language using lpc features. In *Proceedings of the 2013 E-Health and Bioengineering Conference (EHB)*, pages 1–4, Iasi, Romania.

[Fu et al., 2021] Fu, E., Li, X., Yao, Z., Ren, Y., Wu, Y., and Fan, Q. (2021). Personnel emotion recognition model for internet of vehicles security monitoring in community public space. *Eurasip Journal on Advances in Signal Processing*, 2021:81.

[Gers et al., 2000] Gers, F. A., Schmidhuber, J., and Cummins, F. (2000). Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471.

[Goodfellow et al., 2020] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

[Gouveia et al., 2020] Gouveia, C., Tomé, A., Barros, F., Soares, S. C., Vieira, J., and Pinho, P. (2020). Study on the usage feasibility of continuous-wave radar for emotion recognition. *Biomedical Signal Processing and Control*, 58:101835.

[Graves et al., 2006] Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural

networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

[Grosz and Sidner, 1986] Grosz, B. and Sidner, C. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12:175–204.

[Gruber and Jockisch, 2020] Gruber, N. and Jockisch, A. (2020). Are gru cells more specific and lstm cells more sensitive in motive classification of text? *Frontiers in artificial intelligence*, 3:40.

[Guo et al., 2019] Guo, S., Feng, L., Feng, Z.-B., Li, Y.-H., Wang, Y., Liu, S.-L., and Qiao, H. (2019). Multi-view laplacian least squares for human emotion recognition. *Neurocomputing*, 370:78–87.

[Guo et al., 2020] Guo, S., Pei, H., Wu, F., He, Y., and Liu, D. (2020). Modeling of solar field in direct steam generation parabolic trough based on heat transfer mechanism and artificial neural network. *IEEE Access*, 8:78565–78575.

[Hasnul et al., 2021] Hasnul, M., Aziz, N., Alelyani, S., Mohana, M., and Aziz, A. (2021). Electrocardiogram-based emotion recognition systems and their applications in healthcare—a review. *Sensors*, 21:5015.

[Hassan et al., 2021] Hassan, L., Abdel-Nasser, M., Saleh, A., A. Omer, O., and Puig, D. (2021). Efficient stain-aware nuclei segmentation deep learning framework for multi-center histopathological images. *Electronics*, 10(8):954.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Hinton et al., 2012] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., et al. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29:82–97.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Houben et al., 2015] Houben, M., Van Den Noortgate, W., and Kuppens, P. (2015). The relation between short-term emotion dynamics and psychological well-being: A meta-analysis. *Psychological Bulletin*, 141:901.

[Hsu et al., 2021] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

[Idris and Salam, 2015] Idris, I. and Salam, M. (2015). Improved speech emotion classification from spectral coefficient optimization. In *Proceedings of Advances in Machine Learning and Signal Processing: Proceedings of MALSIP 2015*, pages 247–257, Ho Chi Minh City, Vietnam.

[Ilyurek,] Ilyurek. Bsc statistics | data enthusiast | middle east technical university. https://www.linkedin.com/in/ilyurek/.

[Info, 2024] Info, A. P. (2024). Unsupervised learning. Accessed: 2024-05-17.

[Jiang and Fan, 2020] Jiang, J. and Fan, J. A. (2020). Simulator-based training of generative neural networks for the inverse design of metasurfaces. *Nanophotonics*, 9(5):1059–1069.

[Kakouros et al., 2023] Kakouros, S., Stafylakis, T., Mošner, L., and Burget, L. (2023). Speech-based emotion recognition with self-supervised models using attentive channel-wise correlations and label smoothing. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

[Kakuba et al., 2022] Kakuba, S., Poulose, A., and Han, D. S. (2022). Attention-based multi-learning approach for speech emotion recognition with dilated convolution. *IEEE Access*, 10:122302–122313.

[Kartali et al., 2018] Kartali, A., Roglić, M., Barjaktarović, M., Đurić Jovičić, M., and Janković, M. (2018). Real-time algorithms for facial emotion recognition: A comparison of different approaches. In *Proceedings of the 2018 14th Symposium on Neural Networks and Applications (NEUREL)*, pages 1–4, Belgrade, Serbia.

[Kim and Lee, 2023] Kim, J.-Y. and Lee, S.-H. (2023). Coordvit: A novel method of improve vision transformer-based speech emotion recognition using coordinate information concatenate. In *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–4. IEEE.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kong et al., 2017] Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., and Zhang, Y. (2017). Short-term residential load forecasting based on lstm recurrent neural network. *IEEE transactions on smart grid*, 10(1):841–851.

[Koti et al., 2024] Koti, V. M., Murthy, K., Suganya, M., Sarma, M. S., Kumar, G. V. S., and Balamurugan, N. (2024). Speech emotion recognition using extreme machine learning. *EAI Endorsed Transactions on Internet of Things*, 10.

[Krizhevsky et al., 2017] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

[Kusy and Zajdel, 2014] Kusy, M. and Zajdel, R. (2014). Application of reinforcement learning algorithms for the adaptive computation of the smoothing parameter for probabilistic neural network. *IEEE transactions on neural networks and learning systems*, 26(9):2163–2175.

[Kwon et al., 2021] Kwon, S. et al. (2021). Att-net: Enhanced emotion recognition system using lightweight self-attention module. *Applied Soft Computing*, 102:107101.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[Lee, 2019] Lee, S.-W. (2019). The generalization effect for multilingual speech emotion recognition across heterogeneous languages. In *Proceedings of ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5881–5885, Brighton, UK.

[Li and Deng, 2020] Li, S. and Deng, W. (2020). Deep facial expression recognition: A survey. *IEEE Transactions on Affective Computing*, 13:1195–1215.

[Li et al., 2010] Li, X., Li, X., Zheng, X., and Zhang, D. (2010). Emd-teo based speech emotion recognition. In *Proceedings of the Life System Modeling and Intelligent Computing: International Conference on Life System Modeling and Simulation, LSMS 2010, and International Conference on Intelligent Computing for Sustainable Energy and Environment, ICSEE 2010*, pages 180–189.

[Li et al., 2021] Li, Z., Liu, F., Yang, W., Peng, S., and Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019.

[Lin et al., 2022] Lin, T., Wang, Y., Liu, X., and Qiu, X. (2022). A survey of transformers. *AI open*, 3:111–132.

[Mandryk et al., 2006] Mandryk, R., Atkins, M., and Inkpen, K. (2006). A continuous and objective evaluation of emotional experience with interactive play environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1027–1036, Montréal, QC, Canada.

[Martin et al., 2016] Martin, O., Kotsia, I., Macq, B., and Pitas, I. (2016). The enterface'05 audio-visual emotion database. In *Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06)*, page 8, Atlanta, GA, USA.

[Medsker and Jain, 1999] Medsker, L. and Jain, L. C. (1999). *Recurrent neural networks: design and applications*. CRC press.

[Mehrabian and Russell, 1974] Mehrabian, A. and Russell, J. (1974). *An Approach to Environmental Psychology*. The MIT Press, Cambridge, MA, USA.

[Mihalache and Burileanu, 2023] Mihalache, S. and Burileanu, D. (2023). Speech emotion recognition using deep neural networks, transfer learning, and ensemble classification techniques. *Science and Technology*, 26(3-4):375–387.

[Mijwil, 2018] Mijwil, M. (2018). Artificial neural networks advantages and disadvantages. Consulté le 2 avril 2021.

[Mills, 2016] Mills, M. (2016). Artificial intelligence in law: The state of play 2016. *Thomson Reuters Legal executive Institute*.

[Mohammed et al., 2016] Mohammed, M., Khan, M. B., and Bashier, E. B. M. (2016). *Machine learning: algorithms and applications*. Crc Press.

[Mukhamediev et al., 2021] Mukhamediev, R. I., Symagulov, A., Kuchin, Y., Yakunin, K., and Yelis, M. (2021). From classical machine learning to deep neural networks: a simplified scientometric review. *Applied Sciences*, 11(12):5541.

[Mukherjee et al., 2020] Mukherjee, A., Jain, D. K., Goswami, P., Xin, Q., Yang, L., and Rodrigues, J. J. (2020). Back propagation neural network based cluster head identification in mimo sensor networks for intelligent transportation systems. *IEEE Access*, 8:28524–28532.

[Muppidi and Radfar, 2021] Muppidi, A. and Radfar, M. (2021). Speech emotion recognition using quaternion convolutional neural networks. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6309–6313. IEEE.

[Na et al., 2016] Na, W., Feng, F., Zhang, C., and Zhang, Q.-J. (2016). A unified automated parametric modeling algorithm using knowledge-based neural network and $_1$ optimization. *IEEE Transactions on Microwave Theory and Techniques*, 65(3):729–745.

[Nassif et al., 2019] Nassif, A., Shahin, I., Attili, I., Azzeh, M., and Shaalan, K. (2019). Speech recognition using deep neural networks: A systematic review. *IEEE Access*, PP:1–1.

[Nayak et al., 2021] Nayak, S., Nagesh, B., Routray, A., and Sarma, M. (2021). A human-computer interaction framework for emotion recognition through time-series thermal video sequences. *Computers & Electrical Engineering*, 93:107280.

[Nema and Abdul-Kareem, 2018] Nema, B. and Abdul-Kareem, A. (2018). Preprocessing signal for speech emotion recognition. *Journal of Science*, 28:157–165.

[Ogata and Sugano, 1999] Ogata, T. and Sugano, S. (1999). Emotional communication between humans and the autonomous robot which has the emotion model. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, pages 3177–3182, Detroit, MI, USA.

[Oh et al., 2021] Oh, G., Ryu, J., Jeong, E., Yang, J., Hwang, S., Lee, S., and Lim, S. (2021). Drer: Deep learning-based driver's real emotion recognizer. *Sensors*, 21:2166.

[Oikonomidis et al., 2022] Oikonomidis, A., Catal, C., and Kassahun, A. (2022). Hybrid deep learning-based models for crop yield prediction. *Applied artificial intelligence*, 36(1):2031822.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

[Picard, 2000] Picard, R. W. (2000). *Affective computing*. MIT press.

[Plutchik, 2003] Plutchik, R. (2003). *Emotions and Life: Perspectives from Psychology, Biology, and Evolution*. American Psychological Association, Washington, DC, USA.

[Rahmani et al., 2021] Rahmani, A. M., Yousefpoor, E., Yousefpoor, M. S., Mehmood, Z., Haider, A., Hosseinzadeh, M., and Ali Naqvi, R. (2021). Machine learning (ml) in medicine: Review, applications, and challenges. *Mathematics*, 9(22):2970.

[Ranaweera and Mahmoud, 2021] Ranaweera, M. and Mahmoud, Q. H. (2021). Virtual to real-world transfer learning: A systematic review. *Electronics*, 10(12):1491.

[Rani et al., 2020] Rani, P., Verma, S., Nguyen, G. N., et al. (2020). Mitigation of black hole and gray hole attack using swarm inspired algorithm with artificial neural network. *IEEE access*, 8:121755–121764.

[Rattanyu et al., 2010] Rattanyu, K., Ohkura, M., and Mizukawa, M. (2010). Emotion monitoring from physiological signals for service robots in the living space. In *Proceedings of the ICCAS 2010*, pages 580–583, Gyeonggi-do, Republic of Korea.

[Rayas-Sánchez, 2004] Rayas-Sánchez, J. E. (2004). Em-based optimization of microwave circuits using artificial neural networks: The state-of-the-art. *IEEE Transactions on Microwave Theory and Techniques*, 52(1):420–435.

[Rescigno et al., 2020] Rescigno, M., Spezialetti, M., and Rossi, S. (2020). Personalized models for facial emotion recognition through transfer learning. *Multimedia Tools and Applications*, 79(47):35811–35828.

[Rusek et al., 2020] Rusek, K., Suárez-Varela, J., Almasan, P., Barlet-Ros, P., and Cabellos-Aparicio, A. (2020). Routenet: Leveraging graph neural networks for network modeling and optimization in sdn. *IEEE Journal on Selected Areas in Communications*, 38(10):2260–2270.

[Rusydi et al., 2019] Rusydi, M. I., Anandika, A., Rahmadya, B., Fahmy, K., and Rusydi, A. (2019). Implementation of grading method for gambier leaves based on combination of area, perimeter, and image intensity using backpropagation artificial neural network. *Electronics*, 8(11):1308.

[Sahu et al., 2015] Sahu, R. K., Panda, S., and Padhan, S. (2015). A novel hybrid gravitational search and pattern search algorithm for load frequency control of nonlinear power system. *Applied Soft Computing*, 29:310–327.

[Saste and Jagdale, 2017] Saste, S. and Jagdale, S. (2017). Emotion recognition from speech using mfcc and dwt for security system. In *Proceedings of the 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, pages 701–704, Coimbatore, India.

[Scarselli et al., 2008] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

[Schafer and Rabiner, 1975] Schafer, R. and Rabiner, L. (1975). Digital representations of speech signals. *Proceedings of the IEEE*, 63:662–677.

[Schuller, 2018] Schuller, B. (2018). Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends. *Communications of the ACM*, 61:90–99.

[Science, 2024] Science, . D. (2024). Decision trees in machine learning. Accessed: 2024-05-17.

[Seaton, 2021] Seaton, H. (2021). *The construction technology handbook*. John Wiley & Sons.

[Sharma et al., 2017] Sharma, S., Sharma, S., and Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316.

[Singh et al., 2021] Singh, P., Srivastava, R., Rana, K., and Kumar, V. (2021). A multimodal hierarchical approach to speech emotion recognition from audio and text. *Knowledge-Based Systems*, 229:107316.

[Su and Kuo, 2019] Su, Y. and Kuo, C.-C. J. (2019). On extended long short-term memory and dependent bidirectional recurrent neural network. *Neurocomputing*, 356:151–161.

[Suganthi et al., 2015] Suganthi, L., Iniyan, S., and Samuel, A. A. (2015). Applications of fuzzy logic in renewable energy systems–a review. *Renewable and sustainable energy reviews*, 48:585–607.

[Sun et al., 2020] Sun, X., Song, Y., and Wang, M. (2020). Toward sensing emotions with deep visual analysis: A long-term psychological modeling approach. *IEEE Multimedia*, 27:18–27.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

[Tabassum et al., 2014] Tabassum, M., Mathew, K., et al. (2014). A genetic algorithm analysis towards optimization solutions. *International Journal of Digital Information and Wireless Communications (IJDIWC)*, 4(1):124–142.

[Takayama et al., 2000] Takayama, K., Morva, A., Fujikawa, M., Hattori, Y., Obata, Y., and Nagai, T. (2000). Formula optimization of theophylline controlled-release tablet based on artificial neural networks. *Journal of controlled release*, 68(2):175–186.

[Taye, 2023] Taye, M. M. (2023). Understanding of machine learning with deep learning: architectures, workflow, applications and future directions. *Computers*, 12(5):91.

[Tian et al., 2017] Tian, W., Liao, Z., and Zhang, J. (2017). An optimization of artificial neural network model for predicting chlorophyll dynamics. *Ecological Modelling*, 364:42–52.

[Tomè et al., 2016] Tomè, D., Monti, F., Baroffio, L., Bondi, L., Tagliasacchi, M., and Tubaro, S. (2016). Deep convolutional neural networks for pedestrian detection. *Signal processing: image communication*, 47:482–489.

[Town,] Town, D. Database town. https://databasetown.com. Accessed: 2024-05-17.

[Ullah et al., 2023] Ullah, R., Asif, M., Shah, W. A., Anjam, F., Ullah, I., Khurshaid, T., Wuttisittikulkij, L., Shah, S., Ali, S. M., and Alibakhshikenari, M. (2023). Speech emotion recognition using convolution neural networks and multi-head convolutional transformer. *Sensors*, 23(13):6212.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Venugopal, 2019] Venugopal, P. (2019). State-of-health estimation of li-ion batteries in electric vehicle using indrnn under variable load condition. *Energies*, 12(22):4338.

[Weiss et al., 2016] Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):1–40.

[Wijayasingha and Stankovic, 2021] Wijayasingha, L. and Stankovic, J. A. (2021). Robustness to noise for speech emotion classification using cnns and attention mechanisms. *Smart Health*, 19:100165.

[Wong and Sridharan, 2001] Wong, E. and Sridharan, S. (2001). Comparison of linear prediction cepstrum coefficients and mel-frequency cepstrum coefficients for language identification. In *Proceedings of the 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001*, pages 95–98. IEEE.

[Wu et al., 2016] Wu, H., Zhou, Y., Luo, Q., and Basset, M. A. (2016). Training feedforward neural networks using symbiotic organisms search algorithm. *Computational intelligence and neuroscience*, 2016.

[Wu et al., 2020] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

[Xu et al., 2024] Xu, C., Liu, Y., Song, W., Liang, Z., and Chen, X. (2024). A new network structure for speech emotion recognition research. *Sensors*, 24(5):1429.

[Xue et al., 2019] Xue, Y., Tang, T., and Liu, A. X. (2019). Large-scale feedforward neural network optimization by a self-adaptive strategy and parameter based particle swarm optimization. *IEEE Access*, 7:52473–52483.

[You et al., 2006] You, M., Chen, C., Bu, J., Liu, J., and Tao, J. (2006). Emotion recognition from noisy speech. In *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo*, pages 1653–1656.

[Yousefpoor et al., 2021] Yousefpoor, M. S., Yousefpoor, E., Barati, H., Barati, A., Movaghar, A., and Hosseinzadeh, M. (2021). Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review. *Journal of Network and Computer Applications*, 190:103118.

[Zappone et al., 2019] Zappone, A., Di Renzo, M., Debbah, M., Lam, T. T., and Qian, X. (2019). Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks for wireless system optimization. *IEEE Vehicular Technology Magazine*, 14(3):60–69.

[Zepf et al., 2020] Zepf, S., Hernandez, J., Schmitt, A., Minker, W., and Picard, R. (2020). Driver emotion recognition for intelligent vehicles: A survey. *ACM Computing Surveys (CSUR)*, 53:1–30.

[Zhang et al., 2020] Zhang, Z., Cheng, Q. S., Chen, H., and Jiang, F. (2020). An efficient hybrid sampling method for neural network-based microwave component modeling and optimization. *IEEE Microwave and Wireless Components Letters*, 30(7):625–628.

Appendix A

Work environment and development Tools

Introduction

This appendix presents the essential hardware and software tools used to implement our system. It covers key hardware and software components that support computing needs.

Hardware tools

We utilized the following hardware configuration to complete our work:

Google Colab Hardware:

- CPU: 2x Intel(R) Xeon(R) CPU @ 2.30GHz
- RAM du système: 12.7 GB
- RAM du GPU: 15.0 GB
- Disk: 78.2 GB
- GPU: T4

Software tools

Google colaboratory: With Google Colaboratory, also known as Colab, you can create and run Python code directly from your browser. Colab is an as-a-service version of Jupyter Notebook. A free and open-source product of the Jupyter Project is Jupyter Notebook. **Python programming language:** Python is a high-level, interpreted, object-oriented programming language with an emphasis on readability and an easy-to-learn syntax. Python was actually

created with readability in mind, resembling an English language with a strong mathematical component in its syntax.

Used libraries:

- **os:** The os library in Python is a standard library that provides functionality to interact with the operating system in a portable manner. It allows for file and directory operations such as creating, deleting, and manipulating files and directories. The library also enables access to environment variables, management of processes, and execution of shell commands. Additionally, it offers tools for path manipulation to construct, parse, and normalize file paths across different operating systems. The os library is essential for tasks involving file system interaction, environment configuration, and system command execution within Python applications.

- **Torchaudio:** is a specialized library within the PyTorch ecosystem for audio and speech processing. It simplifies the implementation and experimentation with machine learning models by providing a range of tools and functionalities. These include I/O utilities for loading and saving audio files in various formats, pre-built audio transformations like spectrograms and MFCCs, and built-in support for popular audio datasets. Torchaudio's functional API offers low-level control for custom operations, and its seamless integration with other PyTorch modules facilitates end-to-end workflows. This library streamlines the development process for audio-based machine learning applications, from data preprocessing to model training and evaluation.

- **NumPy:** It is a Python programming language module designed to work with matrices or multidimensional arrays and the mathematical operations that these arrays perform. More specifically, this collection of free and open-source software offers, several functions that enable the manipulation of vectors, matrices, and polynomials in addition to the ability to immediately build a table from a file or, conversely, store a table in a file. SciPy is a collection of Python libraries centered on scientific computing, built on top of NumPy.

- **Pandas:** An open-source Python package called Pandas offers strong features for data manipulation and analysis. Because of its effective and adaptable data structures, it is extensively utilized in workflows related to data science and data analysis. The main data structure in the DataFrame, a two-dimensional data structure resembling a table with labeled rows and columns, is created by Pandas. Pandas is a well-liked option for activities like data wrangling, data cleaning, exploratory data analysis, and data visualization because it makes it simple to load, clean, convert, and analyze structured data. Pandas makes working with tabular data easier by offering a wide number of functions and methods that make it simple for users to edit and extract insights from their data.

- **Matplotlib:** Matplotlib is widely recognized as the most popular library for data visualization and exploration. It offers a broad range of tools for creating basic graphs, such

as line charts, scatter plots, histograms, bar charts, and pie charts. Matplotlib serves as the foundation for many other visualization libraries. It is a plotting library specifically designed for the Python programming language and its numerical extension, NumPy. By using Matplotlib, users can visualize patterns, trends, and correlations that might not be detected by simply examining textual data.

Conclusion

The hardware and software tools detailed in this appendix are foundational for creating a conducive and productive environment for software development.

Appendix B

Implementation steps

Introduction

In this annex, we outline the various steps for implementing the speech emotion recognition classification system. These crucial steps include importing necessary libraries, splitting data, creating the HuBERT model, the training process, and final testing.

Importing libraries

First, we will import all the modules needed to train our model. Figure B.1 shows a piece of code that imports the necessary libraries.

```
import os
import pandas as pd
import numpy as np
import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader
import matplotlib.pyplot as plt
```

Figure B.1: Libraries used

Data splitting

We have divided this dataset into three subsets: one for training, one for validation, and one for testing (train/val/test). The division of this data was carried out using the code shown below:

```
BATCH_SIZE = 4
TRAIN_SIZE = 0.8
VAL_SIZE = 0.1

# Create the full dataset
full_dataset = AudioDataset(df, 'path', 'label_class', train_size=TRAIN_SIZE + VAL_SIZE)

# Split the full dataset into training and validation/test sets
train_size = int(len(full_dataset) * TRAIN_SIZE)
val_size = int(len(full_dataset) * VAL_SIZE)
test_size = len(full_dataset) - train_size - val_size
```

Figure B.2: Splitting dataset

The figure shows a snapshot of Python code used for splitting a dataset into training, validation, and test sets. The code uses the AudioDataset class to create a dataset from a DataFrame and then calculates the sizes of the subsets based on the specified proportions (80% for training, 10% for validation, and 10% for testing). This ensures that the dataset is appropriately partitioned for training and evaluating a speech emotion recognition model.

Creating the HuBERT model

To create and initialize the HuBERT (Hidden-Unit BERT) model for speech emotion recognition, we used the following code:

```
model = HubertAudioModel().to(device)
next(model.parameters()).device
```

Figure B.3: HuBERT Model

Training process

As clarified by the following function, ”training” is the phase where a model learns from data to improve its performance. this crucial process for our model, enabling it to learn and optimize its ability to make accurate predictions or classifications based on the provided data.

```
def train_step(model: torch.nn.Module,
               train_loader: torch.utils.data.DataLoader,
               val_loader: torch.utils.data.DataLoader,
               loss_fn: torch.nn.Module,
               optimizer: torch.optim.Optimizer,
               accuracy_fn,
               device: torch.device = device):

    train_loss, train_acc = 0, 0
    val_loss, val_acc = 0, 0

    # Put model into training mode
    model.train()
```

Figure B.4: train step

Test

We have ten percent of our dataset that will be used to assess the caliber of our model. These test results are new data for our model because they were not used in the learning process.

```
def test_step(model: torch.nn.Module, data_loader: torch.utils.data.DataLoader,
              loss_fn: torch.nn.Module, accuracy_fn, device: torch.device = device):
    ### Testing
    # Setup variables for accumulatively adding up loss and accuracy
    test_loss, test_acc = 0, 0

    # Put the model in eval mode
    model.eval()
```

Figure B.5: test step

Here are the test results of our system:

- For RAVDESS dataset

```
model_result = eval_model(model, test_dataloader, loss_fn, accuracy_fn)

model_result

{'model_name': 'HubertAudioModel',
 'model_loss': 0.829942524433136,
 'model_acc': 84.25925925925925}
```

Figure B.6: Test results for the RAVDESS dataset

- For CREMA-D dataset

```
model_result = eval_model(model, test_dataloader, loss_fn, accuracy_fn)
```

```
model_result
```

```
{'model_name': 'HubertAudioModel',
 'model_loss': 1.2699140310287476,
 'model_acc': 70.93373493975903}
```

Figure B.7: Test results for the CREMA-D dataset

Conclusion

These systematic implementation steps, we ensure the successful development and precise evaluation of the HuBERT model for speech emotion recognition classification.